リモートエクスプロイト攻撃を効率的に観測可能なマルウェア動的解析手法の提案

村上 洸介　　藤井 孝好　　吉岡 克成　　松本 勉

横浜国立大学
240-8501 神奈川県横浜市保土ヶ谷区常盤台 79-7
{murakami, fujii}@mlab.jks.ynu.ac.jp, {yoshioka, tsutomu}@ynu.ac.jp

あらまし これまでに我々はマルウェアの感染手法の一つであるリモートエクスプロイト攻撃の観測を行うためのマルウェア動的解析手法を提案している. 当該手法では, 解析環境内の犠牲ホスト上で実行されたマルウェアからの攻撃を, 解析環境内の第二の犠牲ホストに転送することで安全かつ効率的にリモートエクスプロイト攻撃を観測するが, 第二の犠牲ホストが有する脆弱性への攻撃以外は観測できなかった. そこで本論文では, マルウェアからの攻撃を受ける犠牲ホストを複数用意し, 様々なバージョンのサービスを動作させた上で, これらのホストにマルウェアからの攻撃を適切に転送することで, 一回の検体実行で効率的に攻撃を観測可能な手法を提案する.

A Proposal of Malware Sandbox Analysis Method for Efficient Observation of Remote Exploits

Kousuke Murakami　Takayoshi Fujii
Katsunari Yoshioka　Tsutomu Matsumoto

Yokohama National University
79-7 Tokiwadai, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-8501 Japan
{murakami, fujii}@mlab.jks.ynu.ac.jp, {yoshioka, tsutomu}@ynu.ac.jp

Abstract In this paper, we propose a malware sandbox analysis for efficient observation of remote exploits by malware. In the method, a to-be-tested malware sample is executed on the primary victim host in a sandbox, and then packets from the infected host to vulnerable ports such as 135/tcp and 445/tcp are properly directed to each of multiple secondary victim hosts running different versions of OSes and network services. We implemented the proposed method, evaluated it with malware samples in the wild, and confirmed that we can observe more remote exploits using our method.

1　はじめに

近年, インターネット上でウィルスやワーム, ボットといった悪意のあるソフトウェア, いわゆるマルウェアによる, 個人情報の漏洩, スパムメール, フィッシングなどの被害が深刻となっている. マルウェアの感染手法の一つに, 攻撃対象ホスト上のネットワークサービスの脆弱性を突いて権限を奪取するリモートエクスプロイト攻撃がある. DEP (Data Execute Prevention)[9] や ASLR (Address Space Layout Randomization)[7]といった対策技術の導入により, マルウェアの感染手法はWebブラウザやプラグインの脆弱性を突いたドライブバイダウンロード等にシフトしつつあるものの, 2008年末に発見されその後大流行したConfickerや, 2010年に原発制御システムへの感染が確認されたStuxnetといった重要なマルウェアもリモートエクスプロイト攻撃による感染機能を有しており, その対策は現在も重要である.

マルウェアの対策を考える上でマルウェアの挙動を明らかにすることは重要であり, 近年, 解析環境内でマルウェアを実際に動作させ, その挙動を観測するマルウェア動的解析が注目されている. マルウェア動的解析は. 代表的なものとして NORMAN Sandbox[13], CWSandbox[4], Anubis[8]などが挙げられるが, これらのマルウェア動的解析ではマルウェア検体を感染させるために用意するホストは 1 つであるため, リモートエクスプロイト攻撃を観測することが出来ない. そこで我々は, 解析対象のマルウェアを解析環境内の犠牲ホスト上で実行し, 犠牲ホストからの通信のうち, リモートエクスプロイト攻撃を含む可能性が高い通信を, 解析環境内に用意した第二の犠牲ホストに転送することで, 安全かつ効率的に攻撃を観測する手法[5]を提案している. しかしながら, Windows などの OS 上で動作するサービスはそのバージョン毎に異なる脆弱性を持っている場合があり, 異なる脆弱性に対しては攻撃方法も異なる. 一方, 論文[5]の手法では第二の犠牲ホストは単一であったため, 当該ホストが有する脆弱性への攻撃以外は観測することができないという問題があった.

そこで, 本論文では, 攻撃を受ける側の犠牲ホストを複数用意し, これらのホスト上で様々なバージョンのサービスを動作させておき, マルウェアからの攻撃を適切に転送す

ることで，一回の検体実行で効率的にリモートエクスプロイト攻撃を観測可能なマルウェア動的解析手法を提案する．またハニーポット等で収集した実マルウェア検体を実際に解析し，解析結果に対して文献[3]の手法を適用することでシェルコードの検知を行い提案手法の評価をする．

以降，2 章で我々の提案手法とその実装例を説明し，3 章で実マルウェアを用いた評価実験について述べ，4 章でまとめと今後の課題とする．

2 リモートエクスプロイト攻撃を効率的に観測可能なマルウェア動的解析手法の提案

本章ではリモートエクスプロイト攻撃を効率的に観測可能なマルウェア動的解析手法を提案する．まず 2.1 節では提案手法の概要を説明し，2.2 節で提案手法を実装したマルウェア動的解析システムについて説明する．

2.1 提案手法

本節では解析環境内にマルウェアからの攻撃を受ける犠牲ホストを複数設置し，リモートエクスプロイト攻撃を効率的に観測する動的解析手法を提案する．提案手法の概要図を図 1 に示す．

第一犠牲ホスト:第一犠牲ホストはマルウェア検体を実行し，その挙動を観測するためのホストである．第一犠牲ホスト内に監視ツールを予め用意しておくことで，内部挙動の観測も可能である．

アクセスコントローラ:アクセスコントローラは，第一犠牲ホスト上でマルウェアが行う通信を擬似インターネットまたは実インターネットへと転送する役割を持つ．事前に設定されたフィルタリングルールに従い，マルウェアが行う通信のうち，感染活動，スパムメールの送信など，他ホストへの実被害につながる可能性のある通信については擬似インターネットへ転送し，危険性が十分に低いと判断された通信のみ実インターネットへと転送する．アクセスコントローラのフィルタリングルールの設定は，当該検体の動的解析を複数回行い，前述の例に加え，通信量や宛先 IP アドレスの決定方法など複数の基準に基づき危険性の判定を行う，文献[6]の手法の適用を想定している．

擬似インターネット:擬似インターネットは，実インターネット上のサーバ群を模擬することでマルウェアに対してネットワークサービスを提供する．擬似インターネット内部では HTTP や FTP,SMTP,IRC といったプロトコルに対応した簡易サーバが動作しており，犠牲ホストからの要求に従いダミー応答を返すように設定されている．また，擬似インターネット内には後述の第二犠牲ホスト群も設置されている．

第二犠牲ホスト群:第二犠牲ホスト群は擬似インターネット内に設置され，様々なサービスを起動したホストが複数動作しており，アクセスコントローラにより転送されたマルウェアからのリモートエクスプロイト攻撃に応答する．

解析マネージャ:解析マネージャは，動的解析システムの中核として第一犠牲ホスト，第二犠牲ホスト群の OS イメージ管理，解析対象マルウェア管理，アクセスコントローラの設定，擬似インターネット中の各種サービス・ホストの管理と解析結果の出力を行う．

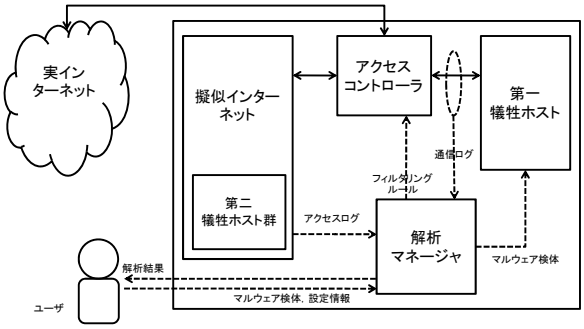


図 1. 提案手法概要図

2.2 実装

提案手法の実装例を図 2 に示す．提案手法を適用した動的解析システムはマシンのスペック上の理由からマシン A とマシン B の 2 台の実マシン上に実装した．第一犠牲ホストは VMware Server v2.0.2 のゲスト OS (Windows SP Professional SP1)を用いた．第一犠牲ホストの他，解析マネージャ，アクセスコントローラ，第二犠牲ホスト群を除いた擬似インターネットについてはマシン A に実装した．また，第二犠牲ホスト群，解析マネージャ，アクセスコントローラについてはマシン B に実装した．以下，各構成要素の実装について述べる．

第一犠牲ホスト:第一犠牲ホストは前述の通り，VMware Server v2.0.2 のゲスト OS により実現した．今回 OS の種類は Windows XP Professional SP1 としたが，OS イメージの差し替えと初期設定を行うことで容易に第一犠牲ホスト上の OS の変更が可能である．解析マネージャにより第一犠牲ホストが起動されると，予め設定されたタスクが Windows のタスクスケジューラにより起動され，マルウェア検体ファイルをホスト OS より SSH 経由でダウンロードし，ゲスト OS 上で自動的に実行する．また近年では，自身が仮想環境上で実行されていることを検知し本来の挙動を示さなくなるようなマルウェアも存在するため[2]，ゲスト OS の NIC の MAC アドレスの変更や，VMware が通信に用いるポートの変更など一定のカモフラージュ処理を行い，これらのマルウェアに対応している．

第二犠牲ホスト群:第二犠牲ホスト群は VMware Server v2.0.2 のゲスト OS により実装した．Windows 2000，Windows XP Pro SP1，Windows 7 はそれぞれ 1 台ずつ仮想マシンを作成し，低対話型ハニーポット nepenthes[1]及び dionaea[10]については Ubuntu 11.04 のゲスト OS 上にインストールした．尚，Ubuntu 11.04 では NIC を二つ用意し，それぞれのインターフェース上で各ハニーポットのサービスが動作するようにした．

ブロードバンドルータ:今回の実装ではマシン A とマシン B の 2 台構成となったため，マシン A とマシン B の間で通信を行うためにマシン A の NIC（以下，ethA とする）とマシン B の NIC（以下，ethB とする）は市販品のブロードバンドルータへ繋がっており，それぞれローカル IP アドレスが振られている．マシン A，B が実インターネットと通信する際にはブロードバンドルータが NAT を行いローカル IP アドレスをグローバル IP アドレスに変換する．

アクセスコントローラと擬似インターネット:アクセスコントローラは Linux のパケットフィルタリングツールである iptables を用いており，マシン A マシン B 両方に設置されている．マシン A のアクセスコントローラでは NAT を行っており，第一犠牲ホストから外部ホストへの接続要求に対して実インターネットへの接続を許可する場合には iptables の PREROUTING チェインで ACCEPT ターゲットを用い，POSTROUTING チェインで MASQUERADE ターゲットを適用することで ethA の IP アドレスへの変換を行い，他ホストと通信できるようにした．一方，第一犠牲ホストからの接続要求に対して実インターネットへの接続を許可せず，擬似インターネット中の各種簡易サーバへ転送する場合には，REDIRECT ターゲットを用いた．

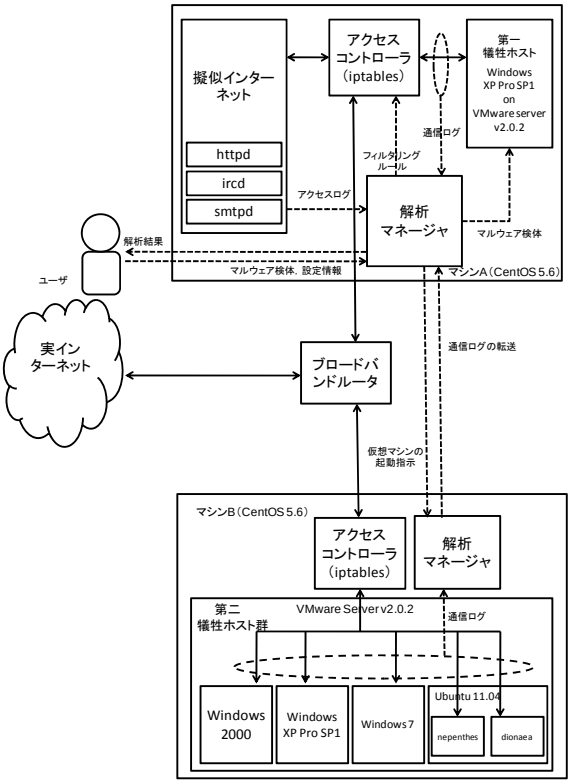


図 2. 実装例

マシン B のアクセスコントローラは第一犠牲ホストから転送された通信を iptables の DNAT ターゲットを用いて第二犠牲ホスト群の各ホストへ転送する．以下に第一犠牲ホストの通信を第二犠牲ホスト群の各ホストへ転送する際の設定について述べる．

① マシン B の ethB には予めエイリアス機能を用いて第二犠牲ホスト群中の各ホスト分の IP アドレスを割り当てておく．今回は 5 つの IP アドレスを割り当てた．

② 第二犠牲ホスト群へ転送する通信は tcp や udp といったプロトコル名，ポート番号により指定し，設定ファイルに保持した．

③ 転送するポートは iptables の PREROUTING チェインで DNAT ターゲットを用いて，宛先を①で割り当てた複数の IP アドレスへと変換する．iptables では DNAT ターゲットの宛先を複数指定すると対象ポートの通信を各セッション毎に宛先 IP アドレスをラウンドロビン的に変換する仕様となっている．

④ ethB に割り当てられた各 IP アドレスは第二犠牲ホスト群の各ホストの IP アドレスと一対一で対応しており，マシン B の iptables の PREROUTING チェインで DNAT ターゲットを用いることで ethB の IP アドレスから各ホストの NIC の IP アドレスへと変換される．

以上の流れで第一犠牲ホストからの通信が第二犠牲ホスト群の各ホストへと転送される．このとき，第一犠牲ホストのマルウェアからは実際のインターネット上のホストへ通信を行っているように見える．また，擬似インターネットを構成する簡易サーバ群については Perl スクリプトにより実装した．

パケット転送の流れを図 3 に示す．

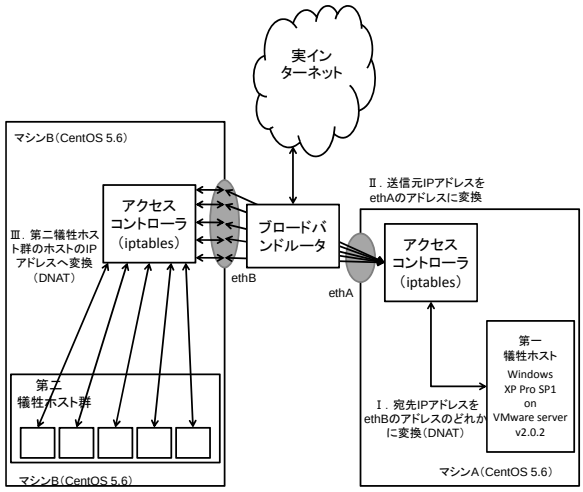


図 3. パケット転送の流れ

解析マネージャ:解析マネージャの動作は Perl スクリプトにより実装した．解析マネージャもまた，マシン A とマシン B 双方に実装されており，解析マネージャは文献[6]で示されている機能の他に，第二犠牲ホスト群の OS イメージ管理や通信ログを scp によりマシン A に転送する役割を持つ．

3 評価実験

提案方式の有効性を検討するため，実現形態の一つである，2.2 節で述べたシステムにより実マルウェアを解析する評価実験を行った．3.1 節では実験方法について説明し，3.2 節では実験結果を示す．さらに 3.3 節では実験結果に対して考察を行う．

3.1 実験方法

実験 1

提案手法の有効性を確認するために，nepenthes で

2007 年 8 月から 2010 年 7 月の間に収集した 4952 検体のマルウェア（以下，検体セット 1）を提案手法を用いて解析を行った．表 1 は解析時の設定項目である．アクセスコントローラのフィルタリングルールは実験の簡略化のため，135/tcp や 445/tcp といったリモートエクスプロイト攻撃の標的とされるポートへの接続を除き，第一犠牲ホストから外部ホストの全てのポートへの接続を許可する設定で解析を行った．さらに解析結果に対して文献[3]の手法を用いてシェルコード検知を行った．

実験 2

dionaea を用いて 2011 年 8 月 17 日に収集した，Conficker[15]として検知される検体 10 検体と，インターネット上より取得した Hiberium[16]，Zotob[17]などのマルウェアからなる検体セット（以下，検体セット 2）を提案手法及び文献[5]の手法（以下，既存手法とする）を用いて解析を行った．尚，既存手法の解析環境中には第二犠牲ホストとして nepnethes が動作している．実験 2 もアクセスコントローラのフィルタリングルールは実験の簡略化のため実験 1 と同様の設定とした．さらに提案手法と既存手法の比較のために，実験 1 と同じく文献[3]の手法を用いてシェルコード検知を行った．両手法の解析時の設定を表 1 にまとめる．

表 1. 評価実験の設定

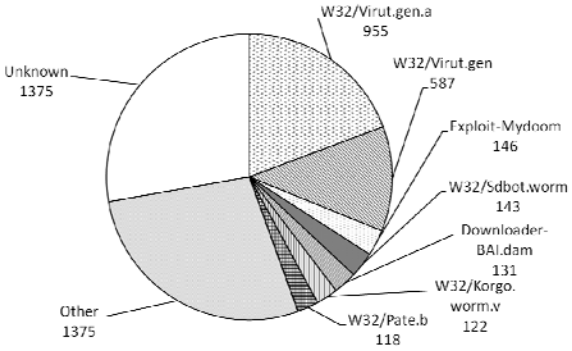
第二犠牲ホスト群へ転送するポート
135/tcp，139/tcp，445/tcp，1025/tcp，5000/tcp
インターネットへの接続を許可するポート
上記ポートと25/tcp，1434/tcpなどを除いたポート
検体実行時間（提案手法）
検体セット1：30秒，検体セット2：60秒，検体セット3：60秒
検体実行時間（既存手法）
検体セット2：60秒，検体セット3：60秒
検体解析回数
1回

3.2 実験結果

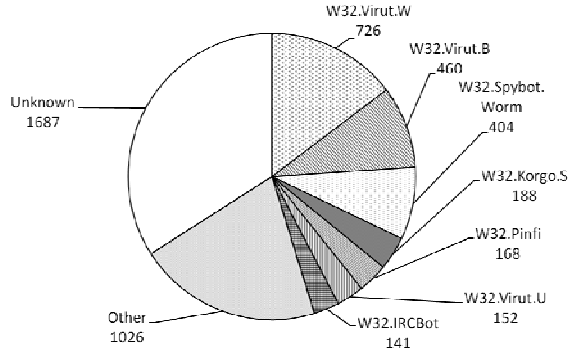
検体セット 1 及び 2 の検体をアンチウィルスソフトによる検知結果を提供するサービスである，VirusTotal[14]のハッシュ値検索を用いて得られた結果を図 4，表 3 に示す．検体セット 1 では Symantec，McAfee どちらの結果も Virut が多数を占めていた．

3.2.1 実験 1 の結果

実験 1 では全 4952 検体中 2168 検体が何らかの通信を第二犠牲ホスト群のホストへ行った．通信の宛先となったポートは 445/tcp が 2126 体の検体から通信を受けており，一番多かった．次いで 139/tcp が 146 体，135/tcp が 14 体であった．ただし，第 2 犠牲ホスト群へ通信を行った 2168 検体中，6 検体は第二犠牲ホスト群の一部のホストへ通信を行っていなかった．この現象については後ほど考察を行う．表 2(a)は実験 1 のシェルコード検知結果である．表 2(a)の各項目は“シェルコードを検知した検体数 / 対象ポートへ通信を行った検体数”を示している．尚，1 検体の解析結果に対していずれか一つの第二犠牲ホスト群のホストへシェルコードを送信していることを検知すれば 1 件としてカウントしている．135/tcp への通信では 14 検体中 10 検体がシェルコードを送信していると判定された．さらに，445/tcp では 2168 検体中 1968 検体がシェルコードを送信していると検知された．上記のシェルコード検知結果を各第二犠牲ホスト群のホスト別に集計したものが表 2(b)である．445/tcp へシェルコードを送信する検体の大多数は Windws XP や nepenthes，dionaea へシェルコードを送信していることがわかる．このような結果から，提案手法は多くの検体に対してシェルコードの観測が可能であることを示していると言える．



McAfee



Symantec

図 4. 検体セット 1 の内訳

3.2.2 実験 2 の結果

まず提案手法の解析結果について述べる．Conficker については全 10 検体が何らかの通信を第二犠牲ホスト群のホストへ行っており，139/tcp，445/tcp に対してそれぞれ 10 検体全てが通信を行っていた．ただし，10 検体中 3 検体は第二犠牲ホスト群の 5 ホストのうち 2 ホストにのみ通信を行っていた．また，残りの 4 検体についても全 4 検体が何らかの通信を第二犠牲ホスト群のホストへ行っており，特に Dasher は 1025/tcp で通信を行っていた．また，Dasher 以外の 3 検体は 445/tcp に対して通信を行っていた．

次に，既存手法で解析した結果では Conficker は全 10 検体が何らかの通信を第二犠牲ホストである nepenthes へ行い，この際の宛先ポートは 445/tcp のみであった．残りの 4 検体の通信の宛先ポートは，同検体を提案手法を用いて解析した際の結果と同様であった．

表 3 は検体セット 2 のシェルコード検知結果である．表 3 の各項目はシェルコードが検知されたポートを示している．また，no と書かれた項目はシェルコードが検知されなかったことを示している．尚，提案手法の結果のうち，前述の，

表 2. 検体セット 1 のシェルコード検知結果

(a)

	135/tcp	139/tcp	445/tcp
検知数	10/14	0/146	1968/2168

(b)

	135/tcp	139/tcp	445/tcp
Windows 2000	9	0	14
Windows XP	8	0	1723
Windows 7	9	0	1
nepenthes	9	0	1916
dionaea	7	0	1910

表 3. 検体セット 2 のシェルコード検知結果

MD5ハッシュ値 McAfee / Symantec	手法	Win2K	WinXP	Win7	nepenthes	dionaea
12b8e29ef86422ae5a0e179e40e9dd2c	提案手法	445/tcp	no	no	no	445/tcp
W32/Conficker.worm.gen.b / W32.Downadup.B	既存手法				no	
21fa69448050de0806e1db9d52478b4b	提案手法	445/tcp	no	no	no	445/tcp
W32/Conficker.worm.gen.a / W32.Downadup.B	既存手法				no	
23fd59ffd64269bb3850e32023d2dab1	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm.gen.a / W32.Downadup.B	既存手法				no	
2f25db0275e8c13bebffaf5993da2f30	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm.gen.a / W32.Downadup.B	既存手法				no	
38202f889d57f7a3dbbd1d55f503d466	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm / W32.Downadup.B	既存手法				no	
3ce10bf31ff9145db3f887a3f9ad6652	提案手法	445/tcp	no	no	no	445/tcp
W32/Conficker.worm.gen.a / W32.Downadup.B	既存手法				no	
7a48406366497b20ffe1b14e35f80683	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm.gen.b / W32.Downadup	既存手法				no	
8358b61628f641bbadede589879adc77	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm.gen.a / W32.Downadup.B	既存手法				no	
b02dee336d0b585d109de4c4e04bbc9b	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm / W32.Downadup.B	既存手法				no	
c54a98f260c63ed5d8ffcbca73ecfbc3	提案手法	445/tcp	445/tcp	no	no	445/tcp
W32/Conficker.worm.gen.b / W32.Downadup.B	既存手法				no	
b71321373bc822e10f07d30a2bd3b804	提案手法	445/tcp	445/tcp	no	445/tcp	445/tcp
W32/Hiberium.gen / W32.Iberio	既存手法				445/tcp	
9c06d095b0d1bdb5aded333f9882c83a	提案手法	445/tcp	445/tcp	no	445/tcp	445/tcp
W32/Zotob.worm.b / W32.Zotob.B	既存手法				445/tcp	
398a1d5ef36e33956bcc293020f13e0b	提案手法	no	no	no	1025/tcp	no
W32/Dasher.worm / W32.Dasher.B	既存手法				1025/tcp	
404283f85c2972fa1e5274dc8bb07783	提案手法	445/tcp	445/tcp	no	no	no
Exploit-MS08-067 / Trojan.Dropper	既存手法				no	

第二犠牲ホスト群ホストの一部へ攻撃が行き渡らなかったケースについては表 3 では no と記載されている. 提案手法を用いて Conficker を解析した結果では Windows 2000, Windows XP, dionaea のホストへの通信でシェルコードが検知され, nepenthes のみが動作している既存手法ではシェルコードが検知されなかった. また, シェルコードを検知したホストでは, Windows 2000 を除いてファイルダウンロードの挙動が観測された. nepenthes への通信からシェルコードが検知されなかった原因としては, Conficker が感染時に悪用する MS08-067[12]の脆弱性に nepenthes が対応していないためなどが考えられる. 逆に dionaea の通信ログからはシェルコード, ファイルダウンロード挙動などが観測された. また, Heberium/Iberio, Zotob の解析結果から提案手法は, Windows 7 以外のすべてのホストの 445/tcp でシェルコードが検知されており, 既存手法でも 445/tcp でシェルコードが検知されていた. しかし, Hiberium/Iberio の結果では dionaea の通信ログからのみ検体ファイルのダウンロードが確認され, その他についてはファイルダウンロードの挙動は見られなかった. また, Zotob の結果ではファイルダウンロードなどの挙動は見られなかった. 特に Heberium/Iberio のシェルコード検知結果は, 両手法でシェルコードが検知されたが, ファイルダウンロードの挙動が観測されたのは提案手法のみという結果になった. これらの原因も Conficker の場合と同じく, Heberium/Iberio が悪用する脆弱性 MS05-039[11]に nepenthes が対応していないことが挙げられる. また, Dasher の結果からは nepenthes のみがシェルコードを受け取っているということがわかった. さらに Exploit-MS08-067/Trojan-Dropper の結果ではシェルコードは Windows 2000, Windows XP の通信ログからのみ検知された. また, 当該検体は Conficker と同じく MS08-067 の脆弱性を悪用するため, 上記脆弱性に対応していない nepenthes の通信ログからはシェルコードが検知されなかった. しかし Conficker の場合と異なり, dionaea の通信ログからはシェルコードが検知されなかった.

3.3 考察

検体の実行時間に関して

前節で述べた通り, 提案手法を用いて検体セット 1, 2 を解析した際には第一犠牲ホストからの通信が第 2 犠牲ホスト群の一部のホストに届かない結果となった. 検体セット 1 のこのような結果となった各検体の通信ログを調査したところ, 第二犠牲ホスト群へ転送されるポートの宛先となっている IP アドレスが各実行時間の中で 5IP アドレス未満であることが分かった. 2.2 節で述べた実装形態では第二犠牲ホスト群のホスト数は 5 ホストであるため, 全てのホストへは通信が届かなかった. 従って, 第二犠牲ホスト群全てにマルウェアからの通信が届いていない場合には検体実行時間を増やして再解析するなどの機能が必要である.

マルウェアの宛先 IP アドレス数について

前述の項目に関連して, マルウェアが第二犠牲ホスト群へリモートエクスプロイト攻撃を行う際に, 1 秒あたりいくつ

の IP アドレスへ攻撃を行うかを，検体セット 1 に関して示す(図 5).

今回使用した検体では 1 秒あたり 20 から 30 のホストへ攻撃を行う検体が最も多かった．一方，1 秒あたり 200 以上のホストに攻撃を行う検体も含まれていた.

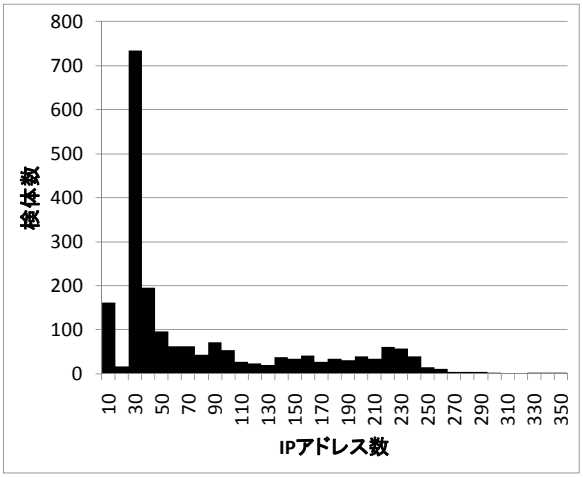


図 5. 1 秒当たりの攻撃先 IP アドレス数

パケットのルーティングに関して

今回の実装では第一の犠牲ホストおよび第二犠牲ホスト群はブロードバンドルータと iptables により構成されるプライベートネットワーク内に位置しているため，外部ホストからこれらのホストへの TCP セッション開始要求は届かない．このため，マルウェアのバックドアへのアクセスなどを観測することができないという問題がある．また，iptables による第二の犠牲ホストへのパケット転送では複数のポート間での関連性は考慮されないため，本来同一ホストの異なるポートに届くはずの通信が，異なるホストに届いてしまい，攻撃が適切に観測できなくなる可能性がある．これらの問題の解決のためには専用のパケット転送ツールが必要である.

4 まとめ

マルウェア動的解析環境中にマルウェアからのリモートエクスプロイト攻撃を受ける犠牲ホストを複数設置することでリモートエクスプロイト攻撃を効率的に観測できる動的解析手法を提案した．また，2 種類の実マルウェア検体セットを提案手法及び，論文[5]の手法を用いて解析を行い，評価を行った．この際，論文[3]の手法で各解析結果に対してシェルコード検知を行い，提案手法と論文[5]の手法で観測したシェルコードの比較を行い，提案手法の有効性を確認した.

今後の課題としては，多数のマルウェア検体を用いた評価を行うことと，3.3 節で述べた，パケットルーティングの改善などが挙げられる.

参考文献

[1] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. C. Freiling, "The Nepenthes Platform: An Efficient Approach to Collect Malware," 9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006), pp. 165 - 184, 2006.

[2] X. Chen, J. Andersen, Z. M. Mao, M. Bailey and J. Nazario, "Towards an Understanding of Anti-virtualization and Anti-debugging Behavior in Modern Malware," The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2008), 2008.

[3] 藤井孝好，吉岡克成，四方順司，松本勉 “エミュレーションに基づくシェルコード検知手法の改善，” マルウェア対策研究人材育成ワークショップ 2010.

[4] C. Willems, T. Holz, and F. Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," Security & Privacy Magazine, IEEE, Volume 5, Issue 2, pp. 32 - 39, 2007.http://www.cwsandbox.org/

[5] K. Yoshioka, D. Inoue, M. Eto, Y. Hoshizawa, H. Nogawa, and K. Nakao, "Malware Sandbox Analysis for Secure Observation of Vulnerability Exploitation," IEICE Trans. Vol. E92D, No.5, pp. 955-966, 2009.

[6] K. Yoshioka and T. Matsumoto, "Multi-pass Malware Sandbox Analysis with Controlled Internet Connection," IEICE Trans. vol.E93-A, no.1, pp. 210-218, 2010.

[7] Address Space Layout Randomization, http://www.symantec.com/avcenter/reference/Address_Space_Layout_Randomization.pdf

[8] Anubis, http://analysis.seclab.tuwien.ac.at/.

[9] Data Execute Prevention, http://support.microsoft.com/kb/875352

[10] dionaea – catches bugs, http://dionaea.carnivore.it/

[11] Microsoft Security Bulletin MS08-067, http://technet.microsoft.com/en-us/security/bulletin/ms08-067

[12] Microsoft Security Bulletin MS05-039, http://technet.microsoft.com/en-us/security/bulletin/ms05-039

[13] NORMAN Sandbox Information Center, http://www.norman.com/microsites/nsic/

[14] VirusTotal, http://www.virustotal.com/jp/

[15] W32.Downadup.B, http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2008-123015-3826-99

[16] W32.Iberio, http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2005-091616-2741-99

[17] W32.Zotob.B, http://www.symantec.com/ja/jp/security_response/writeup.jsp?docid=2005-081415-0741-99