Linux 上で動作するマルウェアを安全に観測可能な マルウェア動的解析手法の提案

田辺 瑠偉† 筒見 拓也† 小出 駿† 牧田 大佑†‡ 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
{tanabe-rui-nv, tsutsumi-takuya-mg, koide-takashi-mx, makita-daisuke-jk}@ynu.jp
{yoshioka, tsutomu}@ynu.ac.jp

‡情報通信研究機構
184-8795 東京都小金井市貫井北町 4-2-1
d.makita@nict.go.jp

あらまし Windowsマシンを攻撃対象としたマルウェアの動的解析については多くの研究が行われているが，近年増加しているLinuxマシンを狙ったマルウェアの動的解析技術については検討が少ない．そこで，本論文ではマルウェア検体を実行する犠牲ホストと犠牲ホストの管理を行う制御ホストを仮想環境内で実現することで，Linux上で動作するマルウェアを安全に解析するため通信制御方式を提案する．また，犠牲ホストをプライベートネットワークやパブリックネットワーク内で実現することで，Linuxマルウェアの動的解析に適したネットワーク構成を提案する．

A Proposal of Malware Sandbox Analysis Method for Safe Observation of Linux Malware

Rui Tanabe† Takuya Tsutsumi† Takashi Koide† Daisuke Makita†‡
Katsunari Yoshioka† Tsutomu Matsumoto†

†Yokohama National University
79-7, Tokiwadai, Hodogaya-ku, Yokohama-shi, Kanagawa 240-8501 JAPAN
{tanabe-rui-nv, tsutsumi-takuya-mg, koide-takashi-mx, makita-daisuke-jk}@ynu.jp
{yoshioka, tsutomu}@ynu.ac.jp
‡National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo 184-8795, JAPAN
d.makita@nict.go.jp

Abstract Recently malware which target Linux machines are increasing. A number of malware sandbox analysis methods have been proposed but not much has been discussed for Linux malware. In this paper we propose a malware analysis method for Linux malware. By implementing two hosts as virtual machines and executing the to-be-analyzed malware sample in one host and controlling the outbound traffic to the Internet by the other we show a safe way to observe Linux malware. Moreover, we show how to properly prepare the network environment in which the malware sample is executed on a host in a private network or that with a global address assigned.

1 はじめに

近年，Linux マシン上で動作するマルウェアが増加している．例えば，Ebury は 2 万 5 千台以上の Linux サーバに感染し，リダイレクトやマルウェアの大量送信に悪用されたことが報告されている[7]．これまで Linux マシンは Windows マシンに比べてマルウェア感染が一般的でなかったことから，セキュリティ対策が十分でないまま使用されている場合や，一部の組み込み機器においては OS やアプリケーションの更新を行ったり，感染の有無を確認するためのユーザインターフェイスの整備が不十分なことから感染に気付かない場合も多い．また，Linux マシンの中にはサーバとして定常稼働しており固定の IP アドレスが割り振られているマシンやルータのような組み込み機器も多く，攻撃用のインフラとして利用価値が高い．このような背景から Linux マシンに感染するマルウェアは今後も増加することが予想される．

マルウェアへの対策を考える上で，その挙動を明らかにすることは重要であり，その方法の一つにマルウェア動的解析がある．マルウェア動的解析では，解析環境内でマルウェア検体を実際に実行し，通信内容や内部挙動を観測する．代表的なマルウェア動的解析システムとして，NORMAN Sandbox [6]，CWSandbox [4]，Anubis [5]などが挙げられる．しかし，これらのマルウェア動的解析システムでは，マルウェアを感染させるために用意する環境は主に Windows であり，Linux を狙うマルウェアの動的解析についてはあまり検討されていない．

そこで，本論文では，仮想環境上にマルウェア検体を実行する犠牲ホスト群と，これらのホストの通信を管理する制御ホストを用意することで，Linux 上で動作するマルウェア検体から外部に対して行われる可能性のある悪性の通信を制御しつつ解析を行う手法を提案する．また，これまで動的解析環境の多くは，犠牲ホストをプライベートネットワーク内に設置し，インターネットとの境界のゲートウェイにおいて通信制御を行うネットワーク環境が多かったが，マルウェアの中にはグローバル IP アドレスが割り振られたサーバ上での活動を前提としたものが存在する[12]．そこで，これらマルウェア検体の解析に適したネットワーク環境と通信制御方法も提案する．

以降，2 章で先行研究について説明し，3 章で手法の提案を行う．そして，4 章で実マルウェア検体を用いた評価実験について説明し，最後に 5 章でまとめと今後の課題について説明する．

2 先行研究

マルウェア動的解析は，実インターネットへの接続性の観点で，完全隔離型とインターネット接続型に分類できる．完全隔離型の解析システムでは，マルウェアからインターネットへのアクセスは許可しないため，外部に悪影響を与えるリスクは低いが，近年のマルウェアは多くはインターネットと接続された環境での動作を前提にしており，本来の動作を観測できない場合が多い．一方，インターネット接続型の解析システムでは多くの情報を収集できるが，外部へ攻撃が流出しないよう適切に制御する必要がある．このため，論文[1]では隔離環境に近い解析環境から解析を開始し，観測された通信の中から危険性が低いと判断された通信を順次インターネットへ接続する手法が提案されている．また，論文[3]では，マルウェアがインターネット上のサーバと行う通信を模擬する疑似クライアントを用いて，マルウェアに対するサーバからの応答を収集し，解析環境にフィードバックする手法が提案されている．

一方，マルウェアの中には感染後にポート待ち受け状態になりサーバとして動作するマルウェアや，最初からサーバマシンを狙うマルウェアが存在する．こうしたマルウェアを解析する場合には，解析環境からインターネットへ接続可能であるとともに，インターネット側から解析環境へ直接接続要求が到達可能である必要がある．論文[2]では，NAT 下の解析環境においてポート待受け状態となったマルウェア検体に対し，NAT を実現しているルータのルーティングテーブルを自動変更することでポート解放環境を実現する手法が提案されている．

このように様々な動的解析手法が提案されているが，Linux マルウェアの中にはグローバル IP アドレスが割り振られたサーバ上での動作を前提とするものも存在し，NAT 下の環境では当該検体が本来の動作をするとは限らない．また，ルータのルーティングテーブルを変更し，インターネット側から送られて

くる接続要求を解析環境に転送する設定であっても，マルウェア検体から発生する送信元 IP アドレスを偽装したパケットはルータで変換，あるいは遮断されてしまうため DRDoS 攻撃（Distributed Reflection Denial-of-Service）などを観測することはできない．そこで，本論文ではこれらの課題を解決するための動的解析システムのネットワーク環境を提案する．

3 Linux マルウェアを安全に観測可能な動的解析手法の提案

本章ではネットワーク環境が異なる 2 つの動的解析手法を提案する．まず初めに，3.1 節で Linux マルウェアを安全に解析するための要件を説明する．そして，3.2 節で NAT 下などのプライベートネットワーク内で解析する手法を提案し，3.3 節でグローバル IP アドレスを割り当てたパブリックネットワーク内で解析する手法を提案する．最後に，3.4 節で提案手法の実装例を説明する．

3.1 Linux マルウェアを安全に解析するための要件

マルウェア動的解析ではマルウェア検体を解析環境内で実行するため，解析システムそのものがマルウェアに感染するリスクや，外部へ感染が拡大するリスクがある．特に，実装の容易性等から解析システム本体の OS として Linux を選択する場合には，検体の取り扱いに注意が必要である．提案手法では，次の 4 つの要件を満たすことで Linux マルウェアを安全に解析する動的解析システムを実現する．

解析システム本体の感染防止：マルウェア検体の保存，管理を行うホスト（以後，制御ホスト）を解析システム上に独立した仮想マシンとして用意し，検体を実行する犠牲ホストへの検体の送信など，検体の取り扱いは全て制御ホストが行うことで，解析システム本体が動作するホストマシン上ではマルウェア検体を一切扱わないようにする．

内部ネットワーク感染の防止：犠牲ホストからシステム内部のホスト(制御ホストやホストマシン等)への通信は全て遮断する．また，通信の制御は犠牲ホスト以外のホストが行うことでマルウェア検体によるアクセス制御ポリシーの書き換えを防ぐ．

外部感染の防止：犠牲ホストから発生する通信のうち，リモートエクスプロイト攻撃やその他の感染拡大活動に利用されるポートへの通信は遮断する．また，特定ホストへの通信量が閾値を超えた場合には一定時間通信の制限をかける．先ほどと同様に，通信の制御は犠牲ホスト以外のホストが行う．

攻撃の検知と通知：犠牲ホストから発生する通信をホストマシン上で監視し，DoS 攻撃等の大量通信が発生した場合には管理者に通知する．また，解析後にアンチウィルスソフトなどによるスキャンを行うことでマルウェア感染を検知する．

3.2 犠牲ホストをプライベートネットワーク内で実現する手法

本節では，マルウェア検体を実行する犠牲ホストがプライベートネットワーク内に存在し，NAT 機器を通じてインターネット接続する環境を説明する．提案手法の概要図を図 1 に示す．また，提案方式の構成要素は次の通りである．

ホストマシン：タスクスケジューラや監視ツール，仮想マシンなど，動的解析システムを実現するためのマシン．

制御ホスト：マルウェア検体の管理や犠牲ホストから発生する通信を制御，観測するためのホスト．制御ホストはホストマシンをデフォルトゲートウェイとするプライベートネットワーク内に実現する．

犠牲ホスト：マルウェア検体を実行するためのホスト．犠牲ホストは容易に差し替えが可能であるとともに，複数用意することで並列解析を行える．また，観測ツールを予め用意しておくことで内部挙動を観測する．犠牲ホストは制御ホストをデフォルトゲートウェイとするプライベートネットワーク内に実現する．

タスクスケジューラ：設定情報をもとに，監視ツールや制御ホスト内の解析マネージャを起動して動的解析を開始する．また，動的解析終了時には犠牲ホストの OS イメージの復元を行う．

監視ツール：制御ホストからホストマシンへ転送されてくる通信（犠牲ホストから発生した通信）を監視し，内部への攻撃を遮断する．また，DoS 攻撃と思われる通信が発生した場合には管理者に通知する．これら以外の通信はインターネットへ転送する．

解析マネージャ：動的解析システムの中核として，マ

ルウェア検体の犠牲ホストへの転送・実行，フィルタリングルールの適用，通信のキャプチャを行う．

検体 DB: マルウェア検体の保存，管理を行う．検体は実行権限が与えられていない状態で保存し，解析の直前に犠牲ホスト上で実行権限を与える．

アクセスコントローラ: 犠牲ホストからの通信をフィルタリングルールに基づいて処理する．フィルタリングルールの設定には，先行研究で述べた論文[1]の手法の適用を想定している．

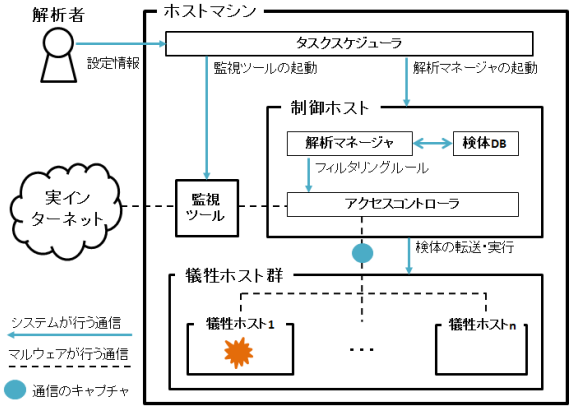

図 1. 提案手法の概要図

なお，犠牲ホストや制御ホストは仮想環境内で実現されるため，解析者のニーズに応じてネットワーク構成は容易に変更可能である．

3.3 犠牲ホストをパブリックネットワーク内で実現する手法

本節では，ホストマシンがパブリックネットワークに存在し，犠牲ホストにホストマシンと同じグローバル IP アドレスが割り当てられた環境を提案する．提案手法の構成要素は 3.2 節と同様である．ホストマシンはルータなどを介さずにインターネットへ直接接続しており，グローバル IP アドレスが割り当てられている．また，インターネット側からホストマシンに届く接続要求パケットは全て犠牲ホストに転送されるよう設定する．犠牲ホストはホストマシンに割り当てられるグローバル IP アドレスと同じ IP アドレスを割り当て，インターネットへの接続を許可する．送信元が偽装されたパケットも閾値を超えない範囲でインターネットへの転送を許可することで送信元 IP アドレス詐称を伴う攻撃も再現可能とする．提案方式における通信の流れを図 3 に示す．内部感染を防止するため，犠牲ホストから制御ホストへの通信，制御ホストからホストマシンへの通信は全て遮断する．また，外部への感染拡大を防止するため，脆弱性攻撃が頻繁に行われる 445/tcp 番ポート等への通信を全て遮断する．

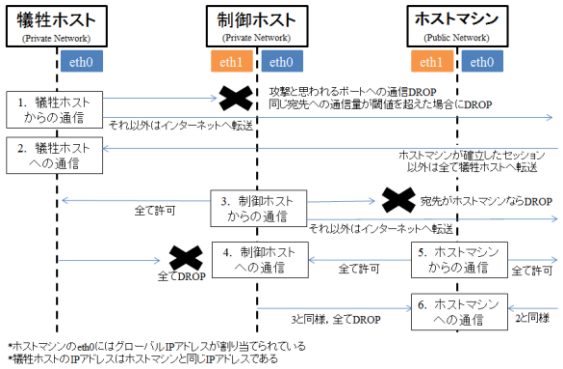

図 2. 提案手法における通信の流れ

3.4 提案手法の実装

本節では提案手法の実装について説明する．提案手法の実装には，論文[1]で提案されているマルチパス動的解析システムの構成を参考に，3.2 節，3.3 節で提案した手法を一台のマシン上に実現する．図 3 に提案手法の実装例を示す．また，各構成要素の実装は次の通りである．

ホストマシン: Ubuntu v12.04.2 を用いて実現する．そして，ルータなどのネットワーク管理機器を介さずにインターネットに直接接続し，Ubuntu の PPPoE 機能を用いてグローバル IP アドレスを直接割り当てた NIC と，制御ホストのデフォルトゲートウェイとなる IP アドレスを割り当てた仮想 NIC を持つ．iptables を用いてこれらのインターフェイス間でパケットが相互転送されるように設定する．設定の詳細は監視ツールの項で述べる．

制御ホスト: Virtual Box v4.2.16 のゲスト OS (Ubuntu v12.04.1)により実現する．そして，ホストマシンと通信可能な IP アドレス割り当てた仮想 NIC と，犠牲ホストのデフォルトゲートウェイとなる IP アドレスを割り当てた仮想 NIC を持ち，iptables を用いてこれらのインターフェイス間でパケットが相互転送されるように設定する．なお，OS のイメージの差し替えは簡単に行う事ができる．

犠牲ホスト: VirtualBox のゲスト OS (CentOS v5.6)により実現する．犠牲ホストには，ホストマシン

と同じグローバル IP アドレスを割り当てた犠牲ホスト A とローカル IP アドレスを割り当てた犠牲ホスト B の 2 種類のホストを用意し，3.2 節で提案した手法を実装する場合にはホスト B を，3.3 節で提案した手法を実装する場合にはホスト A を用いる．

監視ツール: iptables を用いて，犠牲ホストや制御ホストから発生する内部ホストへのパケットを DROP する．これら以外のパケットは POSTROUTING 機能（SNAT）を用いてインターネット上へ転送する．ただし，送信元 IP アドレスが偽装されたパケットについてはそのままインターネット上へ転送する．一方，インターネット側からホストマシンへ送られてくる接続要求パケットは PREROUTING 機能（DNAT）を用いて制御ホストへ転送する．また，tcpdump を用いてホストマシン上を流れるパケットを監視し，通信量が閾値を超えた場合には解析者に通知する．

アクセスコントローラ: iptables を用いて犠牲ホストから発生する脆弱性攻撃などによく使われるポートへのパケットや内部ホストへのパケットを DROP するとともに，hashlimit 機能を用いて特定のホストへの通信量が閾値超えた場合には通信制限をかける．これら以外のパケットは POSTROUTING 機能を用いて，インターネット上へ転送する．ただし，送信元 IP アドレスが偽装されたパケットについては，そのままインターネット上へ転送する（hashlimit 機能は有効）．一方，監視ツールの PREROUTING 機能により転送されてきたパケットについては，アクセスコントローラで再度 PREROUTING を行い，犠牲ホスト A へ転送する．

タスクスケジューラ・解析マネージャ・検体 DB: これらは perl や Linux コマンドにより実現する．

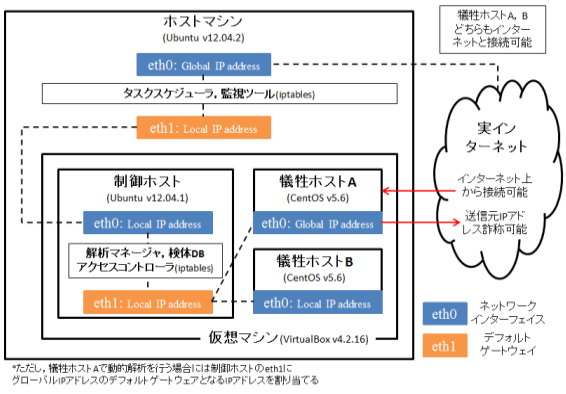

図 3. 提案手法の実装例

4 評価実験

本章では 3 章で提案した手法の有効性を検討するため，提案手法の実現形態の一つである 3.4 節で述べた動的解析システムを用いて Linux マルウェアの動的解析を行った．まず初めに，4.1 節で実験内容を説明し，4.2 節で実験結果をまとめる．そして，4.3 節で考察を行う．

4.1 実験内容

提案手法の有効性を検討するため，Linux マルウェアの動的解析を行った．まず初めに，Web 上で公開されている解析レポートを参考に，DoS 攻撃を行うことが確認されている検体を VirusTotal[8]からダウンロードした．そして，インターネット接続を許可しない環境で動的解析を行い，特徴的な通信を行った Trojan-DDoS.Linux.DNSAmp.a 検体（以後，DNSAmp）に対し，プライベートネットワーク内（3.2 節で提案した手法）で長期間動的解析を行った．

次に，提案手法が他の検体に対しても有効であることを検討するため，VirusTotal からダウンロードした前述の 45 検体と 2014 年 8 月 11 日に新たにダウンロードした 131 検体，さらに，低対話型ハニーポット kippo [9]，glastopf [10]を用いて 2013 年 12 月 30 日から 2014 年 8 月までに収集した 8 検体，合計 184 検体のマルウェア（以後，検体セット）に対してプライベートネットワーク内で動的解析を行った．図 4 に検体セットの Kaspersky による検知結果をまとめる．ただし，*は任意の文字列を表す．

最後に，3.3 節で提案した手法（パブリックネットワーク内で解析する手法）の有効性を検討するため，検体セット内の Net-Worm.Linux.Darlloz.a 検体（以後，Darlloz）の動的解析を行った．表 1 に動的解析実験における設定情報を，表 2 に検体 DNSAmp と検体 Darlloz の情報をまとめる．

表 1. 設定情報

	DNSAmp	Darlloz	検体セット
検体実行時間	30日	6時間	1検体当り5分
実験日	2014/6/27 ~ 7/27	2014/8/24 17:42 ~ 23:42	2014/8/11 ~ 13
検体数	1検体	1検体	184検体
インターネット接続を遮断する通信	135,139,445,1025,3127,3306,3389, 5000,9986番ポートへのudp/tcp通信		
通信制限	特定のホストとの通信は250ppmまでとする		

表 2. 長期動的解析を行った検体

検知名 (Kaspersky)	MD5ハッシュ値
Trojan-DDoS.Linux.DNSAmp.a	84c588c0c63813c436dea8f723e3137a
Net-Worm.Linux.Darlloz.a	00a299fd149939cec860c71224b77209

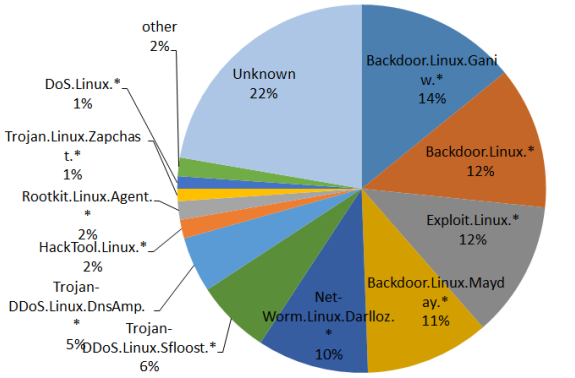

図 4. 検体セットの内訳

4.2 実験結果

DNSAmp の解析結果

DNSAmp（ハッシュ値:84c588c0c63813c436dea8f723e3137a）をプライベートネットワーク内で 30 日間動的解析した．図 5 に解析結果をまとめる．DrWeb の解析レポート[12]によると，当該検体は 37368/tcp 番ポートで C&C サーバと通信を行い，DoS 攻撃の攻撃元として利用される．本実験でも，実行直後に 37368/tcp 番ポートでインターネット上のホストとセッションを確立し，解析が終了するまで定常的に通信を行っていた．通信の中身は暗号化されていたが，通信サーバから当該検体へペイロードサイズが大きいパケットが送信されると，当該検体から DoS 攻撃と思われる大量の通信が発生した（以後，このことを DoS 攻撃事例，または単に事例と呼ぶ）．

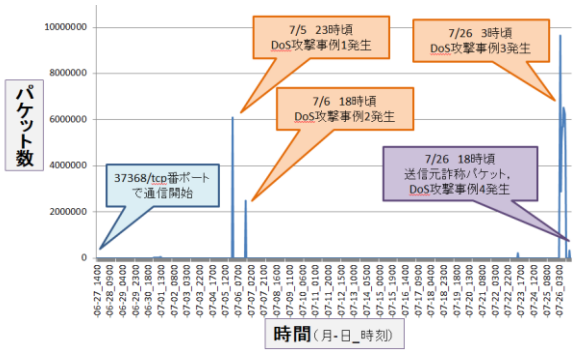

図 5. DNSAmp の通信内容

各事例の通信先 IP アドレスは毎回異なっており，複数のホストへ攻撃を試みた．また，攻撃に使われるポート番号も様々であった．事例 1 は我々が観測した攻撃の中でも単位時間当たりの通信量が最大のものであったが比較的短時間で終了した．一方で，事例 2 では DoS 攻撃が 1 時間以上続いた．事例 3 では，同時に 6 つのホストを攻撃した．各ホストへ約 50 分間，同じ順番で通信を行い，30 分間の休憩をはさんだ後，再び同じ通信と休憩を 5 回繰り返し行った．事例 4 では，攻撃が発生する直前に送信元 IP アドレスを詐称したパケットの送信を試みた．送信元 IP アドレスには 10 万以上の詐称 IP アドレスが利用され，この通信が約 1 分間続いた後，送信元 IP アドレスを詐称せずに同じ宛先 IP アドレスへ大量の通信（事例 4）を行った．送信元 IP アドレスを詐称したパケットの送信理由は様々考えられるが，3.3 節で提案した動的解析システムを用いる事で更なる挙動を観測できる可能性がある．表 4 に各事例の詳細を，表 6 に送信元 IP アドレスを詐称した通信の詳細をまとめる．通信先 IP アドレスは匿名化している．また，通信制限をかけているため，インターネットへ送信されるパケットは 250ppm まであるとともに，送信元 IP アドレスを詐称したパケットは全て DROP した．

表 4. DoS 攻撃と思われる事例

事例	通信先IPアドレス	通信先ポート	継続時間（秒）	pps	Mbps
事例1	110.84.x.x	80/tcp	39	19139	10
	70.39.x.x	99/tcp	682	2900	2.4
	198.100.x.x	21/tcp	322	10455	8.9
事例2	162.212.x.x	21/tcp	4058	616	0.53
事例3	70.39.x.x	53/udp	3000s×5	1回当たり約5400pps	1回当たり約0.14Mbps
	117.25.x.x				
	36.249.x.x				
	121.14.x.x				
	218.66.x.x				
	61.131.x.x				
事例4	123.108.x.x	53/udp	53	5935	0.17

表 6. 送信元詐称パケットの詳細

送信元IPアドレス	103.*.*.*, 188.*.*.*(*は任意の数字)
通信先IPアドレス	123.108.x.x, 8.8.x.x
通信先ポート	53/udp
通信内容	yntrz.com.の名前解決(名前解決の結果は123.108.x.x)
継続時間（秒）	53秒（18:38:29~18:39:22）
パケット数	363,912（6866pps）

検体セットの解析結果

検体セットをプライベートネットワーク内で動的解析した．この結果，77 検体から通信が発生した．通信を行った検体の多くは 10991/tcp 番ポート 36667/tcp 番ポート，37368/tcp 番ポートで C&C サーバと思われるホストと通信を行った．また，中には，58455/tcp 番ポートでネットワークスキャンを行う検体や 80/tcp 番ポートで HTTP 通信を行う検体が存在した．通信を行った検体の多くは，通信先 IP アド

レスを取得するためのドメイン名前解決を行わず，直接 IP アドレスを指定して通信を行った．また，同じ IP アドレスへ通信を行う検体も存在したため，通信先 IP アドレスのリストを内部に保持していることが予想される．表 3 に検体セットの解析結果をまとめる．今回の実験では 184 検体の内，半数以上の検体から通信を観測できなかったが，マルウェアの挙動は不確定であり，複数回解析を行う事で通信を観測できる可能性がある．また，OS のバージョンや種類，ネットワーク環境を変えることで動作する可能性があるため，提案手法の実装条件をかえることでより多くの検体の挙動を観測できることが予想される．

表 3. 通信を行った検体の内訳

Kaspersky 検知名 (*は任意の文字列)	通信を行った検体数/検体数	通信先ポート一覧
Backdoor.Linux.Ganiw.*	24 / 26	25000,28000,30000,36000,36001,36002, 36667,45000,46000,46001/tcp, 53/udp
Net-Worm.Linux.Darlloz.*	14 / 18	80,58455/tcp, 53/udp
Trojan-DDoS.Linux.Sfloost.*	12 / 12	201,205,222,444,555,888,905, 999,1001,1002,1901/tcp, 53/udp
Trojan-DDoS.Linux.DNSAmp.*	9 / 9	8998,9090,37368/tcp 53/udp
Backdoor.Linux.Mayday.*	2 / 19	80,10991,58000/tcp, 53/udp
unknown	16 / 41	80, 888, 902, 1001, 6667, 8998, 9090, 37368/tcp, 53/udp

Darlloz の解析結果

Darlloz（ハッシュ値:00a299fd149939cec860c71224b77209）をパブリックネットワーク内で動的解析した．当該検体は組み込み機器を狙うマルウェアであり，感染すると 58455/tcp 番ポートでバックドアを開く[14]．本実験でも，実行直後から 58455/tcp 番ポートでネットワークスキャンを開始した．約 6 時間の解析で 2700 以上のユニークなホストへスキャンを行い，17 ホストとセッションを確立した．セッションを確立したホストのうち，1 番最初にセッションを確立したホストからは犠牲ホストへデータサイズの大きな通信が複数回発生した．また，観測期間内でスキャンを行ったホストの IP アドレスは x.201.16.0/24～x.201.29.0/24（第一オクテッドは匿名化）であり，前述の検体セット内の Darlloz 検体からも同じ IP アドレスへ通信を行う検体が存在した．このアドレスレンジは AVG による解析レポート[14]でも報告されていることから，Darlloz がスキャンするアドレスは固定されており，そのネットワーク内に存在するホストは，Darlloz 感染ホストのその後の挙動に関わる応答を返す，一種の C&C 機能を提供していることが予想される．犠牲ホストのポート待受け状況を調査したところ 58455/tcp 番ポートでポート待受けを行い，観測期間中にインターネット側から接続要求パケットが届いた．接続要求を試みたホストの IP アドレスは，セッションを確立したホストの 1 つであった．このため，Darlloz にはネットワークスキャンによる Darlloz 感染ホストの探索機能と感染ホスト同士が相互に通信し合うことで自身の更新や情報の共有を行う一種の P2P ネットワークが構成されている可能性がある．

4.3 考察

提案手法の有効性

提案手法を用いて Linux マルウェアの動的解析を行った．この結果，様々なマルウェア検体の挙動を観測することができた．特に，パブリックネットワーク内で解析した場合には，インターネット上からの接続要求パケットを観測することができ，プライベートネットワーク内では観測できないパケットを観測することができた．しかし，解析環境に届くパケットが必ずしも実行した検体と関係があるとは限らない．事実，観測期間中に 22/tcp や 445/tcp ポート宛のスキャンが観測された．また，解析環境が 1 つのグローバル IP アドレスを占有してしまうのも難点である．一方，プライベートネットワーク内での解析の場合，犠牲ホストを複数用意することで，一度に多数のマルウェア検体を解析することができる．また，ネットワーク構成を柔軟に変更することできる．ただし，どちらの手法でも犠牲ホスト上に外部に公開するサービスを立ち上げていないため，サーバマシンを狙う検体の解析には更なる工夫が必要である．

提案手法の安全性

提案手法が Linux マルウェアの挙動を安全に観測できること検証するため，FPROT[11]が公開している Linux マシン用の AV ソフトを用いて，ホストマシンのマルウェア感染の有無を調べた．4 章の実験の後，本 AV ソフトによる検査を行ったが，ホストマシンからマルウェア感染は検知されなかった．また，4 章の実験結果を分析したところ，内部ホスト宛の通信や DoS 攻撃と思われる通信は適切に遮断・制限されていた．外部へ感染拡大活動を行うマルウェアについても，アクセスコントローラのフィルタリングルールの作成には論文[1]の手法を用いるため，遮断できることが予想される．

DRDoS 攻撃を行うマルウェアの観測

Linux を狙うマルウェアも C&C 通信や DoS 攻撃機能など，Windows を狙うマルウェアと同様の機能を持っていることがわかった．一方で，近年 DRDoS 攻撃による被害が増加している．DRDoS 攻撃は送信元 IP アドレスを詐称したパケットをリフレクタに送信することで，詐称 IP アドレスへ DoS 攻撃を行う攻撃手法である．DRDoS 攻撃を行うマルウェアが様々報告されているが，DRDoS 攻撃の観測にはインターネット上に送信元 IP アドレスを詐称したパケットを送信可能な解析環境を用意する必要があり，3.3 節で提案した手法が有効である．

5 まとめと今後の課題

Linux マルウェアをプライベートネットワークとパブリックネットワーク内で安全に解析可能な動的解析手法を提案した．評価実験の結果，多数のマルウェア検体をから様々な挙動を観測することができた．

今後の課題としては，3.3 節で提案した手法を用いて DRDoS 攻撃を行うマルウェア検体を解析するとともに，最初からサーバ化しているマシンを狙うマルウェア検体の挙動を観測するためのよりリアリティのある解析環境の構築やマルウェア検体が行う通信挙動だけでなく内部挙動も観測可能な解析環境を構築することである．

謝辞

本研究の一部は，総務省情報通信分野における研究開発委託／国際連携によるサイバー攻撃の予知技術の研究開発／サイバー攻撃情報とマルウェア実体の突合分析技術／類似判定に関する研究開発により行われた．

参考文献

[1] K . Yoshioka and T . Matsumoto , “Multi-pass Malware Sandbox Analysis with Controlled Internet Connection, ” IEICE Transactions on Fundamental of Electronics Communicationsand Computer Sciences . Volume E93-A, No1, pp. 210-218, 2010.

[2] 鉄 穎，吉岡 克成，松本 勉，“マルウェアのポート待ち受け状態を考慮した並列動的解析環境のネットワーク制御,”コンピュータセキュリティシンポジウム 2013(CSS2013) CD-ROM 文集, 3B2-3, 2013.

[3] 笠間 貴弘，吉岡 克成，松本 勉，山形 昌也 衛藤 将史，中尾 康二，“疑似クライアントを用いたサーバ応答蓄積型マルウェア動的解析システム,” コンピュータセキュリティシンポジウム 2009(MWS2009) CD-ROM 論文集, A7-2, 2009.

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

[5] Anubis,
https://anubis.iseclab.org,
last visited 2014/08/20

[6] NORMAN Sandbox,
http://www.norman.com/index,
last visited 2014/08/20.

[7] ESET, “Operation Windigo”
http://www.welivesecurity.com/wp-content/uploads/2014/03/operation_windigo.pdf ,
last visited 2014/08/24.

[8] VirusTotal,
http://www.virustotal.com/jp/,
last visited 2014/08/20.

[9] Kippo,
https://github.com/desaster/kippo,
last visited 2014/08/20.

[10] Glastopf,
http://glastopf.org/, last visited 2014/08/20

[11] F-PROT,
http://www.f-prot.com, last visited 2014/08/20

[12] DDoS Trojans attack,
http://news.drweb.com/show/?i=5760&lng=en&c=9, last visited 2014/08/21.

[13] ITMedia, “Apache を狙う新たなマルウェア出現，悪質サイトにリダイレクト,”
http://www.itmedia.co.jp/enterprise/articles/1305/01/news025.html, last visited 2014/08/24.

[14] AVG, “Linux.Aidra vs Linux.Darlloz: War of the Worms,”
http://blogs.avg.com/news-threats/war-of-the-worms/, last visited 2014/08/24.