半透性仮想インターネットによるマルウェアの動的解析

青木 一史† 川古谷 裕平† 岩村 誠†‡ 伊藤光恭†

†NTT 情報流通プラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11

{aoki.kazufumi,kawakoya.yuhei,iwamura.makoto,itoh.mitsutaka}@lab.ntt.co.jp

‡ 早稲田大学
169-8555 東京都新宿区大久保 3-4-1

iwamura@muraoka.info.waseda.ac.jp

あらまし マルウェアの挙動を解析する手法に動的解析があるが，これまでは閉環境での解析に終始していたため，十分な解析結果が得られていなかった．本稿では，外部との半透過的な通信を実現する仮想インターネット環境で動的解析を行う Botnet Watcher を用い，CCC DATAset 2009 検体の解析を行った．また，動的解析環境が，実行されるコード量にどのような影響を与えるかについて調査した．解析結果から，閉環境での解析に比べ，Botnet Watcher での解析の方がより多くの接続先や，追加プログラムを収集できることを示した．また，ボットのような外部との通信により動作が変わるマルウェアの場合，開環境で解析した方が実行されるコード量が増加することを示した．

Malware behavior analysis using semipermeable network

Kazufumi Aoki† Yuhei Kawakoya† Makoto Iwamura†‡ Mitsutaka Itoh†

†NTT Information Sharing Platform Laboratories
3-9-11 Midori-Cho, Musashino-Shi, Tokyo, 180-8585 Japan

{aoki.kazufumi,kawakoya.yuhei,iwamura.makoto,itoh.mitsutaka}@lab.ntt.co.jp

‡Waseda University
3-4-1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555 Japan

iwamura@muraoka.info.waseda.ac.jp

Abstract Dynamic analyses which execute malwares by the isolated environment cannot obtain an enough result. In this paper, we analyze CCC DATAset 2009 malwares by Botnet Watcher, which use semipermeable virtual network. And we investigate amount of code which executed in isolated environment and semipermeable environment. Results shows that Botnet Watcher can obtain many destinations, additional execution binaries and can increase amount of executed code.

1 はじめに

近年，ボットをはじめとするマルウェアが猛威を振るっている．これらの脅威を抑えるには，マルウェアを解析し，その挙動に応じた適切な対策を講じる必要がある．

膨大な数のマルウェアが出現している現状から，マルウェアを実際に動作させて挙動を分析

する動的解析が注目されている．動的解析は，完全に隔離された環境で行われるものと，実インターネットへの接続が許可された環境で行われるものとに分けられる．前者は，実インターネットへの接続が完全に絶たれているため，解析における安全性を高く保つことができるが，ボットやシーケンシャルマルウェア [1] のように，外部との通信によってはじめて脅威が露見するようなマルウェアについては，十分な解析結果を得ることができない．一方後者は，実インターネットとの接続を許可することで，隔離環境における解析よりも多くの情報を収集できるものの，解析時の安全性には十分配慮する必要がある．

また，動的解析の場合，マルウェアのコードを逆アセンブル等により分析する静的解析とは異なり，動作した部分の挙動しか把握できない．そのため，動的解析でマルウェアの挙動をどの程度把握できたのかは，実行されたコード量に依存すると考えられる．しかしながら，動的解析で実行されたコード量についての調査はこれまで行われていないため，動的解析自体がどの程度解析に有効であるかが明らかにされていない．

本稿では，このような事態を鑑みて，特定の通信のみ外部に通過させる，半透性仮想インターネット環境上でマルウェアの動的解析を実現する Botnet Watcher[2] の実装を行い，CCC DATAset 2009 検体 [3] によりその評価を行った．評価では，閉環境での解析と，開環境での解析とで得られる結果の違いに着目した．さらに，ステルスデバッガ [4] を利用し，動的解析における実行済みコード領域について調査を行った．

2 BotnetWatcher: 半透性仮想インターネットにおける動的解析システム

BotnetWatcher は，実インターネット上の攻撃者からの指令に応じて挙動が変化するボットや，シーケンシャルマルウェアによる追加プログラムのダウンロードといった挙動を追跡するための動的解析システムである．Botnet Watcher の全体概要図を図 1 に示す．Botnet Watcher は

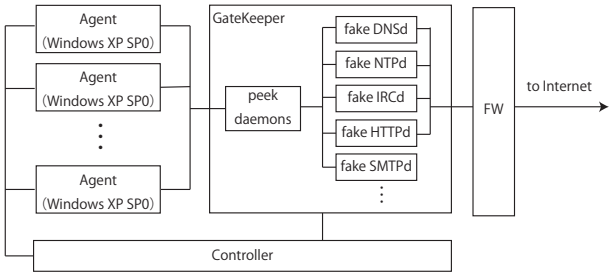


図 1: Botnet Watcher 概要図

Agent / GateKeeper / Controller により構成される．以下，各々について説明する．

2.1 Agent

Agent は，マルウェアを実際に動作させるための環境である．Agent は VMware[5] 上で動作しており，OS として Windows XP（パッチ未適用）をインストールしている．マルウェアによっては，仮想マシン上での動作を検知し，正しく動作しなくなるものが存在する．そこで，Agent には多くのマルウェアが利用する仮想 OS 検知手法である，下記手法を回避するための仕組みを組み込んだ．

- Backdoor I/O ポートアクセス [6]
- VMware 固有の MAC アドレス [6]
- 仮想環境と実環境での SIDT 命令などの戻り値の差異 [7]

2.2 GateKeeper

GateKeeper は，Agent と実インターネットとの間の通信を仲介するモジュールである．GateKeeper では，Agent で実行されたマルウェアからの通信を終端し，実インターネットに通信させるか否かの判断を行う．Agent からの通信は，まず peek daemons で受け取り，ペイロードを解析し，プロトコルの識別を行う．プロトコル識別では，HTTP / SMTP / IRC / DNS / NTP のいずれかであるかを調べている．GateKeeper 内にはこれらのプロトコル毎に処理するための擬

似サーバが用意されており，プロトコルの識別ができた場合には，対応する擬似サーバで外部に通信させるか否かの判断を行う．C&C サーバとの通信や，HTTP によるファイルダウンロードの通信であると判断された場合，擬似サーバは Agent から受け取ったデータを実インターネットに送信し，実インターネットから受信したデータはそのまま Agent に転送する．外部への通信が許可されない場合，擬似応答を生成し，Agent に送信する．プロトコルの識別ができなかった場合には，その旨をログに出力し，Agent に対してレスポンスは送信しない．

2.3 Controller

Controller は，Agent 及び GateKeeper の管理を行う．Agent に対しては，主に解析対象検体の送信と，解析完了時にシステムを解析前の状態にロールバックする作業を行う．また，GateKeeper に対しては，解析完了時にシステムのロールバック作業と，挙動解析ログの収集を行う．なお，Controller では，複数の Agent 及び GateKeeper の操作が可能である．

3 実験

3.1 実験環境

前述した，Botnet Watcher の評価を行うにあたり，研究用データセット CCC DATAset 2009 検体を利用した．実験では，データセットとして提供された 10 検体の挙動解析と，解析時に実行されたコード領域について分析した．以下，検体挙動解析と実行済みコード領域分析の実験環境について説明する．

3.1.1 検体挙動解析

挙動解析は，外部から完全に隔離された閉環境での動的解析と，Botnet Watcher による挙動解析を行い，両者の結果を比較した．閉環境での動的解析は，Matrix Daemons[8] により行った．Matrix Daemons は，隔離環境に構築した仮想インターネット空間内で動的解析を行うためのシステムである．なお，Matrix Daemons には種々の擬似サーバを用意しているが，TCP の通信についてはセッションを確立した後，その通信に対する応答は行っていない．

Matrix Daemons 及び Botnet Watcher では，1 つの検体につき 3 分間動作させて挙動解析を行った．なお，Botnet Watcher については，2009 年 8 月 19 日に解析を行った．

3.1.2 実行済みコード領域の分析

動的解析時に実行されたコード領域について，Botnet Watcher と Matrix Daemons の二つの動的解析環境で調査した．実行済みコード領域は，Agent をステルスデバッガ上で動作させ，EIP をトレースすることで取得した．ステルスデバッガは，仮想マシンを利用してゲスト OS の外側からデバッグ機能を提供するツールである．これにより，マルウェアが持つ種々のアンチデバッグ機能を回避しながらトレースを行うことができる．また，単純にマルウェアにデバッガでアタッチしてトレースする場合，コンテキストスイッチが多発することで動作が重くなってしまうが，ステルスデバッガの場合は一命令毎に例外をハンドリングする必要がないため，高速なトレースを実現することができる．また，実行されたコード領域がオリジナルコードに占める割合を求めるにあたり，[9] の手法によりオリジナルコードを取得した．

3.2 検体解析結果

Matrix Daemons 及び Botnet Watcher で得られたプロトコル別接続先数について，表 1 に示す．表中のかっこ内の数字は，DNS の列では，問い合わせた FQDN の数を表しており，その他の列では，IP アドレスを直接指定してアクセスしたホストの数を表している．また，各検体を識別するために，ID として検体の SHA1 ハッシュの先頭から 4 文字目まで用いた．

HTTP での通信を行ったものについて，1D23，CD91 の検体が IP アドレスを直接指定して行った通信については，いずれも WebDAV へのアクセスを試みるスキャンであった．それ以外の

表 1: Botnet Watcher 及び Matrix Daemons の解析で得られた接続先数（MD: MatrixDaemons, BW: BotnetWatcher）.

	TCP						UDP					
	IRC		HTTP		UNKNOWN		DNS		NTP		UNKNOWN	
ID	MD	BW	MD	BW	MD	BW	MD	BW	MD	BW	MD	BW
1D23	-	-	3	34 (32)	-	172 (172)	1 (3)	1 (5)	-	-	-	-
393F	-	-	3	5	-	7,451 (7,451)	1 (3)	1 (5)	-	-	-	-
68AC	-	-	1	1	-	-	1 (2)	1 (2)	1	1	145 (145)	114 (114)
7190	1	1	-	-	-	3,395 (3,395)	1 (1)	1 (1)	-	-	-	-
84E9	-	-	3	5	-	4,566 (4,566)	1 (3)	1 (8)	-	-	-	-
CD91	-	-	44 (44)	73 (73)	93 (93)	87 (87)	-	-	-	-	6 (6)	- -
D493	1	-	-	-	-	-	1 (1)	1 (1)	-	-	-	-
DF75	-	-	12 (12)	11 (11)	-	-	-	-	-	-	-	-
F8C1	-	-	2	1	-	-	1 (2)	1 (2)	-	-	-	-
FDF3	-	-	1	1	-	-	1 (1)	1 (1)	-	-	-	-

HTTP による通信は，実行プログラムのダウンロードや別サイトへの転送，バイナリデータの受信といった通信に利用されていた．

IRC での通信を行ったものについて，7190 の検体は，Botnet Watcher での解析時に実際に C&C サーバに接続し，近接に対するスキャンを行う命令を受けていた．これにより，Matrix Daemons での解析時には見られていなかった TCP 通信が多数発生している．

また，問い合わせた FQDN の数について比較すると，Matrix Daemons の解析結果に比べ，Botnet Watcher での解析結果の方が多数取得しているのがわかる．同様に，HTTP や IRC での通信先数についても，ほとんどの場合，Botnet Watcher の方が多く取得できている．図 2 は，Botnet Watcher での解析で得られたマルウェアと問い合わせがあった FQDN との関係を表している．1D23，393F，84E9 の検体については，共通の FQDN に対する問い合わせが行われている．共通する問い合わせ先に対するアクセスを詳細に分析したところ，同一のホストから同一のファイルをダウンロードし，実行していた．つまり，これらのマルウェアにはダウンローダとして共通の挙動が存在するといえる．なお，共通してダウンロードしたファイルを静的解析したところ，実行すると apple[snip].com にアクセスするプログラムであった．

1D23，7190，84E9，CD91 の検体に見られるプロトコル判別ができなかった TCP 通信について，宛て先ポートが 135 番，139 番，445

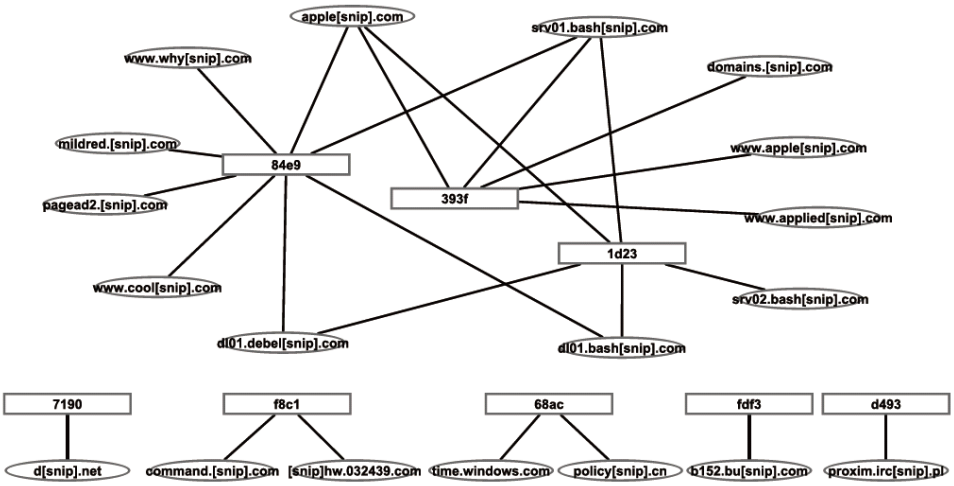


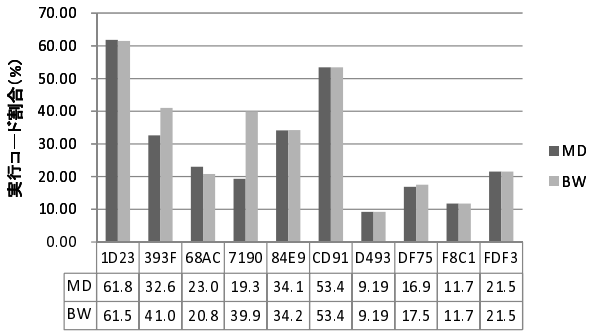
図 2: マルウェアと問い合わせた FQDN の関係（四角：マルウェアのハッシュ値，楕円：問い合わせた FQDN）

番であったことから，これらの通信はいずれもマルウェアの感染活動によるものだと考えられる．また，プロトコル判別ができなかった UDP 通信のうち，68AC の検体についてはデータ長が 25 バイトで，全く同じデータを送出していた．プロトコル判別ができなかった通信については，外部に透過させていないため，今回の動的解析では 25 バイトの UDP の通信がどのような意図で行われたものかを特定することはできなかった．

以上の結果から，Botnet Watcher での解析では，閉環境での解析に比べ，多くの通信先を収集でき，また，シーケンシャルマルウェアがダウンロードする追加ファイルのダウンロードを行うことができている．一方で，プロトコルの判別ができなかった通信については，安全性とのトレードオフから，外部に透過させていないため，実インターネットからどのような応答が返ってくるかを把握できなかった．このように，外部に通過させる通信の選び方は，解析精度に直結するため，外部との通信を許可すべき通信の選び方には課題があるといえる．

3.3 実行済みコード領域

CCC DATAset 2009 検体を，Matrix Daemons 及び Botnet Watcher で動的解析した際



	1D23	393F	68AC	7190	84E9	CD91	D493	DF75	F8C1	FDF3
MD	61.8	32.6	23.0	19.3	34.1	53.4	9.19	16.9	11.7	21.5
BW	61.5	41.0	20.8	39.9	34.2	53.4	9.19	17.5	11.7	21.5

図 3: オリジナルコードに対する実行済みコード領域の割合（MD: Matrix Daemons, BW: Botnet Watcher）

に実行されたコード領域のオリジナルコードに占める割合を図 3 に示す．

図 3 より，393F や 7190 の検体では，Matrix Daemons での解析で実行されたコード量に比べ，Botnet Watcher での解析で実行されたコード量が増加しているのがわかる．これは，外部との通信が成功した場合に実行される領域があるためと考えられる．7190 の検体は，表 1 の結果から，IRC を利用するボットであると考えられるが，Botnet Watcher での解析では，C&C サーバに接続し，指令を受け取ることで感染活動を開始している．この挙動をつかさどるコー

ド領域は，外部からの指令を契機として実行されるので，閉環境での解析だけでは実行するのは難しい．

一方で，CD91 の検体のように，閉環境・開環境のどちらで解析しても実行されるコード領域に差が見られないものが存在する．実行されるコード量に大きな差が見られない原因としては，感染活動が主としてみられるワームや，ダウンロードの機能しか具備していないような検体では，外部からの応答を期待した動作がコーディングされていないことが挙げられる．そのため，表 1 の結果でワームのような挙動が閉環境・開環境の両方で得られていた CD91 の検体では，Matrix Daemon で解析した場合と，Botnet Watcher で解析した場合とで，実行済みコード領域の差が見られなかった．ただし，Botnet Watcher が外部へ許可していない通信でマルウェアが外部からの応答を期待しているようなものが存在する場合にも，Matrix Daemons で実行した場合と差が出ない可能性もある．

また，68AC の検体のように，閉環境の方が実行される領域が大きくなる場合も存在する．この理由としては，外部に通信できない場合の挙動が別途コーディングされているためだと考えられる．

以上のように，外部との通信により挙動が変化する検体の場合には，Botnet Watcher のような環境で動的解析することで，検体のより多くの部分について解析結果を得られることがわかる．

4 まとめ

本稿では，CCC DATAset 2009 検体により，開環境動的解析システムである Botnet Watcher の評価を行った．検体解析に関して，閉環境での動的解析では困難であったボットやシーケンシャルマルウェアの挙動を追跡できることを確認し，さらに，DNS で問い合わせが行われる複数のドメインを閉環境での解析よりも多く収集できることを確認した．また，動的解析時に実行されるコード領域について，閉環境及び開環境で調査を行い，環境によりどの程度実行されるコード領域に差が現れるのかを明らかにした．

参考文献

[1] 独立行政法人 情報処理推進機構セキュリティセンター．近年の標的型攻撃に関する調査研究．http://www.ipa.go.jp/security/fy19/reports/sequential/, 2008.

[2] 青木一史, 岩村誠, 伊藤光恭． 半透性仮想ネットワークを用いたボットの動的解析手法の提案．電子情報通信学会 2008 総合大会, 2008.

[3] 畑田充弘, 仲津留勇, 寺田真敏, 篠田陽一. マルウェア対策のための研究用データセットとワークショップを通じた研究成果の共有. マルウェア対策研究人材育成ワークショップ 2009, 2009.

[4] 川古谷裕平, 岩村誠, 伊藤光恭． ステルスデバッガを利用したマルウェア解析手法の提案．マルウェア対策研究人材育成ワークショップ 2008, 2008.

[5] VMware． Vmware cloud computing with virtualization, green it, virtual machine & servers. http://www.vmware.com/, 2009.

[6] Oudot Laurent. Countering attack deception techniques. *PacSec 2004 Tokyo*, 2004.

[7] Joanna Rutkowska. Red pill... or how to detect vmm using (almost) one cpu instruction. http://www.invisiblethings.org/papers/redpill.html, 2004.

[8] 青木一史, 川古谷裕平, 秋山満昭, 岩村誠, 針生剛男, 伊藤光恭. 能動的攻撃と受動的攻撃に関する調査および考察．情報処理学会論文誌, Vol. 50, No. 9, 2009.

[9] 岩村誠, 伊藤光恭, 村岡洋一. 隠れマルコフモデルに基づく新規逆アセンブル手法．電子情報通信学会 2008 総合大会, 2008.