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

笠間 貴弘† 吉岡 克成† 松本 勉† 山形 昌也‡§ 衛藤 将史§ 中尾 康二§

† 横浜国立大学
‡ NEC 共通基盤ソフトウェア研究所
§ 独立行政法人情報通信研究機構

あらまし 近年のマルウェアはC&Cサーバ等を通じて攻撃者に制御されるものが多いため，マルウェアを解析環境内で実行してその挙動を観測・分析するマルウェア動的解析では，これらのサーバの挙動に注目し，その変化が解析対象のマルウェアの挙動にどのような影響を与えるかを分析する必要がある．しかし，解析環境内で実際にマルウェアを長時間実行して解析を行うことは，解析時間の増加と，解析中のマルウェアが解析環境外部に攻撃を行うリスクの増加を意味する．そこで本稿では，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，動的解析結果から自動生成し，これらをインターネットに接続した別マシン上で並列動作させることで，マルウェアに対するサーバからの応答を継続的かつ効率的に収集する方法を提案する．また，収集された応答を解析環境内にフィードバックすることでサーバの挙動の変化に応じたマルウェアの挙動を観測可能とする．

Malware Sandbox Analysis System with Accumulated Server Responses Collected by Dummy Clients

Takahiro Kasama† Katsunari Yoshioka† Tsutomu Matsumoto†
Masaya Yamagata‡ Masashi Eto§ Koji Nakao§

† Yokohama National University
‡ Common Platform Software Research Laboratories, NEC Corp
§ National Institute of Information and Communications

Abstract Many recent malware change their behaviors according to those of remote servers they interact, such as Command and Control (C&C) servers and file servers. Thus, it is important to carefully observe the changes of behaviors of these servers when malware sandbox analysis, in which malware sample is actually executed in a testing environment (i.e. sandbox) to observe its behavior, is conducted. However, keeping the malware active in an Internet-connected sandbox for a long time to observe its interactions with these servers involves a high risk that their attack may exit the sandbox as well as a high operational cost. In this paper, we propose a new malware sandbox analysis method in which we utilize a dummy client that interacts with the servers instead of a malware sample. The dummy client is automatically created from traffic log observed by sandbox analysis. Server responses collected by the dummy clients are then fed to the executed malware in the sandbox to observe its corresponding reactions.

1 はじめに

重要ファイルの流出，個人情報漏洩，フィッシングといったインターネット上の脅威は後を絶たず，それらの脅威におけるウイルス・ワーム・ボットといったマルウェアの役割は非常に大きい．これらマルウェアによる脅威に対して，マルウェアの収集・解析を通じて，その特性を明らかにし，対策方法を導き出すために，マルウェア解析に関する様々な研究開発が進められている．

マルウェアの解析手法は，大きく分けて，静的解析と動的解析に分類される．静的解析とは，マルウェアの実行可能コードに対してリバースエンジニアリング等の技術を適用することで，マルウェアの機能や構造を解析する手法である．しかし，近年のマルウェアはパッキングと呼ばれる圧縮・難読化技術等を用いているため，静的解析においてはまず，パッキングを解除する必要がある．また一般に，静的解析では解析者にアセンブラ言語や OS に関する高度な知識が要求される．

一方，動的解析は，解析対象のマルウェアを実際に解析環境内で実行し，その挙動を観測・分析する解析手法であるため，パッキングの影響を受けないという利点がある．また，動的解析は静的解析に比べて自動化が容易であるため，増加傾向にあるマルウェアに対する有効な対策として注目されている．

近年のマルウェアの多くは，インターネット上の C&C(Command and Control)サーバやファイルサーバ等の各種サーバから，特定のデータを受信し，その受信データに応じて挙動を変化させるという特徴をもつ．これらのサーバは攻撃者によって管理されており，攻撃者はサーバの挙動の制御を通じて，マルウェア自体を制御する．そのため，同一の検体を異なる時期に動的解析すると，マルウェアの挙動に大きな変化が現れることが報告されている[7]．解析結果が変化する要因としては，マルウェア自体の挙動の変化，アクセス先ドメイン名の DNS 名前解決結果の変化，サーバからの応答の変化等がある．そのため，動的解析では，これらのサーバの挙動にも注目し，その変化が解析対象のマルウェアの挙動に及ぼす影響を分析する必要がある．

動的解析においてこれらサーバの挙動変化の影響を分析する単純な方法としては，サンドボックス内のマルウェアと実インターネットとの通信を許可する解析環境内で実際にマルウェアを長時間実行して

解析を行う方法が考えられる．しかし，マルウェアを長時間実行することは，解析時間の増加および，解析中のマルウェアが解析環境外部に攻撃を行うリスクの増加を意味する．

そこで本稿では，マルウェアがインターネット上のサーバと行う通信を動的解析により観測し，当該通信を模擬する擬似クライアントを自動生成した上で，これらのクライアントを動作させることで，マルウェアに対するサーバからの応答を継続的かつ効率的に収集する方法を提案する．さらに，収集された応答を解析環境内の疑似サーバにフィードバックすることでサーバの挙動の変化に応じたマルウェアの挙動を観測可能とする．

本稿では，CCC DATAset 2009[1]の攻撃通信データから疑似クライアントを自動生成し，これらを用いて提案方式の評価実験を行った．その結果，疑似クライアントを継続的に動作させることで，マルウェアを長時間動作させることなく，マルウェアのアクセス先サーバからの応答の変化を観測できた．さらに疑似クライアントで収集したサーバ応答を解析環境内の疑似インターネットからマルウェアに対して送信することにより，各サーバ応答に対応したマルウェアの挙動を観測できた．

まず，2 章では関連研究について述べ， 3 章で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を提案する．4 章でCCC DATAset 2009 を用いた評価実験について説明し，5 章で考察を行う．そして最後に 6 章でまとめを述べる．

2　関連研究

マルウェア動的解析に関する既存研究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の代表的なものとしては，Norman Sandbox[10]が挙げられる．完全隔離型では，マルウェアから実インターネットへのアクセスを許可しない．しかし，近年のマルウェアの大半はインターネット接続環境での動作を前提に作成されており，隔離された環境では本来の挙動を示さない場合が多い．そこで，Norman Sandbox では，ネットワーク環境をエミュレートすることで，マルウェアに対して疑似的なインターネット環境を提供している．論文[3,5,6]でも同様に完全隔離型の動的解析手法が提案されており，Norman Sandbox と同様に疑似インターネット環境を用いている．これら疑似インターネット環境を利用した解析手法では，いかに実際のインターネット環境を模擬出来るかが課題になってくるが，多様化するマルウェアの外部サーバとの通信に完全に対応することは困難といえる．

一方，インターネット接続型の解析システムの代表的なものとしては，Anubis[9]や CWSandbox[8]が挙げられる．インターネット接続型ではマルウェアから実インターネットへのアクセスを許可するが，外部へ攻撃が流出しないようにフィルタリングを行うのが一般的といえる．しかし，未知のマルウェアを対象とする以上，完全に攻撃を検知・遮断することは困難である．また，隔離環境に近い解析環境で解析を開始し，観測されたマルウェアの通信の中から危険性が低いと判断された通信に関しては実インターネットへの接続を順次許可し，繰り返し解析を行う，自動マルチパス解析によるマルウェア動的解析システムが提案されている[7].

以上のように様々な動的解析手法が提案されているが，前章で述べたような，マルウェアが通信を行うサーバの挙動の変化がマルウェアの挙動に及ぼす影響を分析する場合，完全隔離型では，そもそもサーバの挙動の変化を観測することが出来ない．一方，インターネット接続型であっても，実際にサーバの挙動に変化が現れるまで解析環境内でマルウェアを動かし続ける必要があり，解析時間や外部への攻撃流出のリスクの増加をもたらす．そこで本研究では，上記の課題を解決するための提案を行う．

3 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

本章では，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を提案する．まず 3.1 節では，マルウェア動的解析の評価指標と要件を説明する．3.2 節では，前節で示された要件を満たす方式として提案手法を説明する．3.3 節では，提案システムの実装を説明する．

2.1　マルウェア動的解析の評価指標と要件

以下にマルウェア動的解析の評価指標を挙げる．

Observability : 動的解析によりマルウェアの様々な挙動を観測できる度合を Observability と呼ぶ．観測対象となる挙動は解析の目的に依存して異なるが，我々は，マルウェアの多様な挙動をできる限り詳細に観測・分析することを目標とする．

Efficiency : マルウェアの挙動を安定的かつ効率的に観測できる度合を Efficiency と呼ぶ．解析に要する時間や自動化の可能性も重要な評価指標と考えられる．

Containment : 解析環境自体がマルウェアに感染したり，解析環境の外部に攻撃が流出することなく，安全に解析を行える度合を Containment と呼ぶ．

本提案手法ではマルウェアが通信を行うインターネット上のサーバの挙動の変化を観測し，サーバ応答に応じたマルウェアの挙動を観測可能とすることで Observability の向上を目指す．また，疑似クライアントを用いることで，マルウェアを長期間動作させる方法に比べて Efficiency の向上が期待できる．

2.2　提案手法

本節では，提案手法について説明する．提案手法の概要を図 1 に示す．図 1 において，実線はマルウェア及び疑似クライアントの通信を示し，破線はシステム制御のための通信を示す．提案方式は以下の 7 つの構成要素からなる．

犠牲ホスト：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．

アクセスコントローラ：アクセスコントローラは犠牲ホストにおいて実行されたマルウェアからの通信を疑似インターネット内のダミーサーバに適切に転送する役割を持つ．

疑似インターネット：疑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．この際，データスプールに蓄積されたサーバ応答情報の中から，解析者によって指定された時刻に応じたサーバ応答を返信する．

解析マネージャ：解析マネージャは，犠牲ホストのOSイメージ管理，検査対象マルウェア管理，犠牲ホストからの挙動ログの受信・保管，挙動ログの解析，解析結果の出力を担う．また受信した通信挙動ログを疑似クライアント生成器に送信する．

疑似クライアント生成器：通信挙動ログを基に疑似クライアントを生成する．

擬似クライアント：擬似クライアントは，マルウェアの通信挙動を模擬することで，インターネット上のサーバと通信を行い，その結果受信したデータをデータスプールに保存する．

データスプール：データスプールは，擬似クライアントによって収集したサーバ応答データを蓄積する．データスプールに蓄積させたデータは，擬似インターネットによって読み込まれ，マルウェアからの通信に対して返答される．

解析の流れ：以下に解析の流れを示す．

マルウェア解析及び疑似クライアント生成

①解析対象のマルウェア検体，設定情報(犠牲ホストのOS，マルウェア実行時間，サーバ応答の収集時刻など)を入力すると解析マネージャは，設定情報に従い，犠牲ホスト・擬似インターネットの設定変更を行い，犠牲ホストを起動する．

②犠牲ホストは解析マネージャからマルウェア検体をダウンロードし，これを実行する．マルウェアの全ての通信はアクセスコントローラを介して疑似インターネットに転送される．

③犠牲ホストは設定されたマルウェア実行時間が経過すると収集した内部挙動ログおよび通信挙動ログを解析マネージャに転送する．

④解析マネージャは犠牲ホストからのログ転送が完了すると犠牲ホストを停止し，OSイメージを復元する．また，解析マネージャはログの解析を行い，解析結果を出力するとともに，通信挙動ログを疑似クライアント生成器に転送する．

⑤疑似クライアント生成器は，通信挙動ログを受け取ると，マルウェアから外部への通信のうち，危険性の低い通信を判定し，その通信を模擬する疑似クライアントを生成する．

擬似クライアントによるデータ収集

①擬似クライアントを実インターネットへ接続したマシン上で実行する．

②擬似クライアントは，マルウェアの通信を模擬することでマルウェアに対するサーバからの応答を収集し，保存する．データの保存においては，検索を容易にするため，アクセス先サーバのドメイン名・ポート，プロトコル，データ収集時刻等をラベル付けして保存される．

上記の擬似クライアントによるデータ収集は設定に従い，定期的かつ継続的に行われ，収集されたデータは逐次，データスプールに蓄積される．そして，マルウェアを解析する際には，設定情報としてデータ収集時刻を指定することで，指定時刻における実インターネット上のサーバの挙動を擬似インターネットで模擬し，指定時刻でのサーバの挙動に応じたマルウェアの挙動の解析が可能になる．

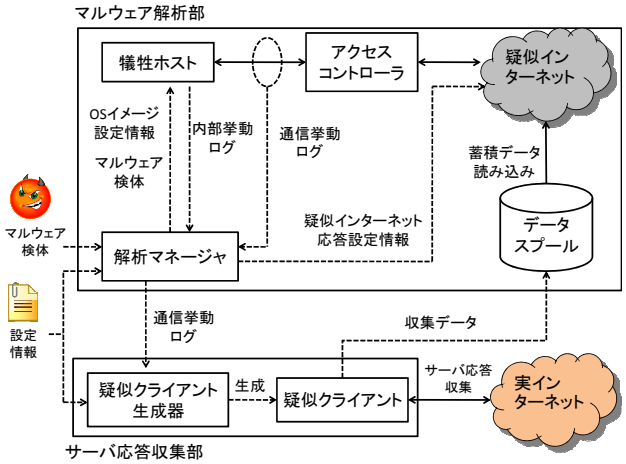


図 1：提案システムの概要図

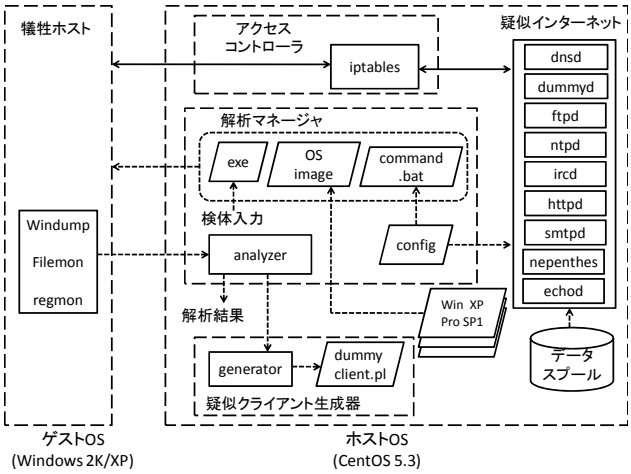


図 2：提案システムの全体図

2.3 実装

本節では提案システムの実装について説明する．当該システム実装においてはマルチパス動的解析システム[7]の構成を基に実装を行った．当該システムの全体図を図 2 に示す．まず，3.3.1 節において当該システムと構成要素の実装を説明し，3.3.2 節において，疑似クライアントの生成方法について述べる．

2.3.1 構成要素

当該システムは，疑似クライアントとその実行環

境を除いて，1 台の実マシン上に実装した．犠牲ホストには VMware Server 1.0.6 のゲスト OS を用い，ホスト OS(CentOS5.3)上にその他の構成要素を実装した．以下，各構成要素を説明する．

犠牲ホスト：VMware Server の OS イメージとしては Windows XP Professional SP1，Windows XP professional SP2, Windows 2000 Professional の 3 種類を用意した．犠牲ホストは起動時に自動的にホスト OS に SSH 接続し，解析マネージャが作成した Windows Batch ファイル command.bat をダウンロードし，実行するよう設定することでホスト OS 側から犠牲ホストに対して任意の命令を実行させられるようにした．

アクセスコントローラ：アクセスコントローラは，パケットフィルタリングツール iptables により実装した．具体的には，マルウェアから通信の全てを PREROUTING チェインにおいて，宛先 IP アドレスを REDIRECT ターゲットによりホスト OS の IP アドレスに書き換えることで，ホスト OS 上のサーバ群に転送されるようにした．その際に，特定の範囲の宛先 IP アドレスへの通信については，宛先ポートを宛先 IP アドレスと対応づけた番号へ書き換えることで，ダミーサーバが REDIRECT 前の宛先 IP アドレスを判別できるようにした．さらに，いずれのサーバの待ち受けポートにも一致しない場合には，echo サーバの待ち受けポートへ転送されるようにした．

疑似インターネット：疑似インターネットは，ホスト OS 上で起動するサーバプログラム群により実現した．まず，DNS サーバとして，マルウェアからの正引きクエリに対して，参照ドメイン名のサーバからの応答データが，データスプール内に存在した場合は特定の範囲の IP アドレスを返答し，存在しない場合はそれ以外の範囲の IP アドレスを返答する，簡易サーバを Perl により実装した．なお，名前解決の際は自動的に逆引きレコードも生成されるようにし，また，DNS サーバによって返答される IP アドレスは，解析中は一意のものとなるようにした．

次に，マルウェアに対して蓄積データを返答するためのダミーサーバを Perl により実装した．ダミーサーバは，マルウェアからのアクセスがあると，宛先 IP アドレスからドメイン名を逆引きする．その後取得したドメイン名と設定された時刻情報に一致するデータをデータスプールより検索し，発見した場合はそのデータを返答する．一致するデータがない場合は echo サーバとして動作する．

その他の疑似サーバについては，HTTP サーバとして Apache2.2.3, NTP サーバとして CentOS 標準の NTP デーモンを用いた．SMTP，IRC，FTP については，各プロトコルにおける初期的なインタラクションのみを模擬する簡易版サーバプログラムを Perl により実装した．さらに犠牲ホスト上のマルウェアが他ホストへの感染活動を行う際の感染対象として低対話型ハニーポット Nepenthes v2.0 を用いた．加えて，これらのサーバプログラムが使用していないポートへのアクセスに対応するために echo サーバを用意した．

解析マネージャ：解析マネージャは config ファイルの内容を読み取り，設定内容に従って，疑似インターネットの設定変更を行い，犠牲ホストの設定を行うための Windows Batch ファイル command.bat を生成する．この command.bat は犠牲ホストの起動時に自動的にホスト OS からダウンロードされて疑似ホスト上で実行されるため，ゲスト OS 側から疑似ホストの設定処理を行うことが可能である．

疑似クライアント生成器：疑似クライアント生成器は，Perl と C 言語によって実装した．疑似クライアント生成器は，通信挙動ログを解析し，危険性の低い通信を判定し，その通信を模擬する疑似クライアントを生成する．

疑似クライアント生成器及び疑似クライアントの実装方法については，どの程度忠実に模擬を行うかによって様々な実装方法が考えられる．マルウェア毎に通信挙動を模擬し，サーバ応答を蓄積すれば，各マルウェアに対応するサーバ応答の取得が可能になる．しかしその一方で，異なる検体が同一サーバに同一クエリを用いてアクセスする場合は，データスプールには重複したデータが保存され，データ量が増大する．

そこで今回の実装では，各マルウェアの通信挙動の模擬ではなく，解析済みの全てのマルウェアの通信挙動をまとめた上で，1 つの疑似クライアントによりサーバ応答を収集する方式を実装した．その際，異なるマルウェアによる同一クエリ等を除去し，疑似クライアントによって模擬する通信量を減らすことで，データ収集を効率化した．なお，Linux 環境上で動作することを想定し，疑似クライアントは Perl スクリプトとした．

2.3.2 疑似クライアント生成方法

疑似クライアントの生成は以下の手順で行われる．
①通信ログを TCP/UDP セッション毎に分割する．
②各セッションに対して以下の基準で危険通信を判断し除去する．

禁止ポートの除外：config で設定した禁止ポートへのセッションは除外する．

セッション開始者判定：セッションの開始パケットの MAC アドレスを調べ，当該セッションが犠牲ホストにより開始されたことを確認する．

送信元 IP アドレス詐称判定：送信元 IP アドレスが偽装されている場合は，除外する．

DNS 名前解決判定：宛先 IP アドレスが DNS 名前解決済みでない場合は除外する．

DoS 攻撃判定：同一宛先 IP アドレスとポートに対して，閾値以上のセッションがある場合は除外する．

③上記の基準を通過した各セッションに対して，サーバのドメイン名，宛先ポート，プロトコルを記憶し，当該セッションにおける送信データを送信順が判別できるように，パケット単位で抽出・結合する．
④抽出した送信データをマルウェアの代わりに送信するスクリプト(疑似クライアント)を生成する．

なお，疑似クライアントによるデータ送信間隔や，データ受信の待機時間は config 内で指定することで

生成時に任意の設定時間が設定できるようにした．また，IRC 通信においては，サーバから自動的に送信される接続確認の PING に対して PONG を自動的に返信するようにした．

3　評価実験

本章では，提案方式の有効性検証のために，CCC DATAset 2009 の攻撃通信データを用いて評価実験を行った結果を示す．

3.1　CCC DATAset 2009

CCC DATAset 2009 はサイバークリーンセンター(CCC)[1]で収集したマルウェア観測データであり，マルウェア検体(ハッシュ値)，攻撃通信データ，攻撃元データから構成されたデータ群である．今回は，そのうちの攻撃通信データを用いて，提案手法の評価実験を行った．攻撃通信データの概要を表 1 に示す．

表 1：攻撃通信データの概要

観測対象	ハニーポット 2 台のパケットキャプチャデータ(pcap ファイル)
データ収集日	2009 年 3 月 13・14 日の 2 日間，
ネットワーク環境	ISP-A，FTTH，動的 IP アドレス
OS 環境	Windows XP SP1，Windows2000

3.2　実験方法

攻撃通信データの中には，ハニーポットに感染した検体が行う通信挙動も含まれている．そこで今回の実験では攻撃通信データを動的解析によって得た通信挙動ログとみなして以下の実験を行った．

事前調査：まず攻撃通信データを疑似クライアント生成器に入力し，マルウェアがインターネット上のサーバと行う通信を抽出した．その上で，抽出したドメイン名に対して，2009/07/20 22:00 から 2009/08/21 22:00 の間，1 時間おきに DNS 名前解決を行い，その応答の変化を調査した．

実験 1：抽出した通信を疑似クライアントにより模擬し，サーバからの応答を継続的に収集した．収集期間は 2009/07/23 から 2009/08/22 とし，疑似クライアントによるサーバ接続は約 6 時間毎に行った．

実験 2：実験 1 で収集したサーバ応答を解析環境に蓄積した上でマルウェア動的解析を行った．

3.3　実験結果

事前調査：攻撃通信データ(2 日分の pcap ファイル)からマルウェアがインターネット上のサーバと行う通信を抽出したところ，19 種類のドメイン名のサーバに対して，計 22 回の TCP セッションが抽出できた．抽出したセッションの送信データを人手で確認したところ，HTTP の GET 要求と IRC プロトコルによる C&C 通信と見られる通信だった．次に，抽出された各ドメイン名について，2009/07/20 22:00～2009/08/21 22:00 の間，1 時間おきに DNS 名前解決を行い，IP アドレスが得られたドメイン名の数を図 3 に示す．

図 3 から，アクセスの時期によって，ドメイン名の名前解決の可否が変化することがわかる．なお，19 個のドメイン名の内，5 つについては常に名前解決ができなかった．7 つについては常に同じ IP アドレスが返信され，残りの 7 つは，名前解決の可否または取得できた IP アドレスに変化が見られた．

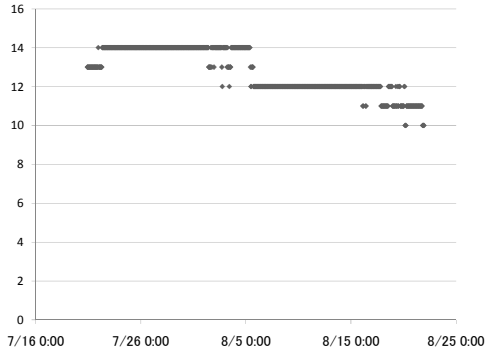
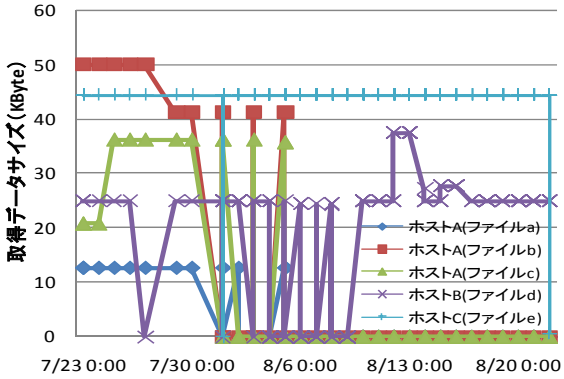
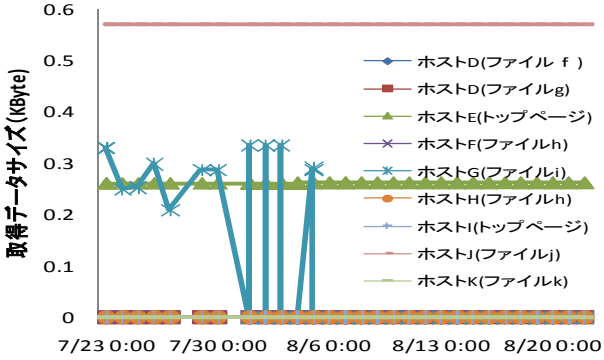


図 3：DNS 名前解決可能なドメイン名数

実験 1：疑似クライアントによって収集したサーバ応答の内，HTTP の GET 要求の模擬によって収集出来たファイルサイズ(ヘッダサイズも含む)の推移を図 4 に示す．なお，視認性のため，取得サイズによって図は 2 つに分けている．



(a)　収集ファイルサイズが 10KB 以上



(b)収集ファイルサイズが 10KB 未満

図 4：疑似クライアントによる収集出来たデータサイズの推移(HTTP の GET 要求)

図 4 のとおり，多くのサーバについては，同一のファイル要求を行っても，その時期によって送られてくるファイルに変化が表れた．収集データが得られない原因としては，DNS 名前解決が出来ない場合や，サーバの応答が無い場合などがある．

実験 2：最後に実験 2 の結果として，疑似クライアントにより収集したサーバ応答をマルウェアに送信した際のマルウェアの挙動の解析結果を示す．なお，説明中のドメイン名等は一部匿名化している．

解析例 1: 攻撃通信データから，あるホスト A の特定ポートで動作する IRC サーバとの C&C 通信を抽出した．そこで疑似クライアントによるホスト A への継続的アクセスを行った結果，毎回，アクセスを行った直後に，“!get http://put.xxx.pl/prepare.exe”のような URL らしき文字列を含むメッセージが送られ，アクセスの時期によってこの文字列が変化することがわかった．次に，当該ホストの当該ポートに対してアクセスを行う検体(MD5 値: 344e310851797df39bd38506bb77f43f シマンテック名: W32.IRCbot)の動的解析時に，疑似クライアントにより当該サーバから収集した応答を用いて解析を行ったところ，当該メッセージ受信後に!get 以下の URL のドメイン名を名前解決し，そのホストの 80/tcp に対して， HTTP GET 要求を送信することがわかった．

解析例 2: 攻撃通信データから，あるホスト B の特定ポートで動作する IRC サーバとの C&C 通信を抽出した．そこで，疑似クライアントによるホスト B への継続的アクセスを行った結果, 当該ホストから IRC チャネルを通じて約 10 分毎に文字列“#las6 :* ipscan s.s.s.s dcom2 -s”を含むメッセージが送信されることがわかった．次に，当該ホストの当該ポートにアクセスを行う検体(MD5 値: c648588ef0cbe00845e33b51ff538b53 シマンテック名: W32.Trats!inf)の動的解析時に，疑似クライアントにより収集したサーバ応答を用いて解析を行った．その結果，サーバ応答を送信しない場合，検体はホスト B の 8080/tcp にアクセスし，IRC メッセージを送信した後は特に通信挙動を示さなかったのに対して，サーバ応答を返信した場合は，検体は別のドメイン名の名前解決を行い，当該ホストに対してファイル rs3.exe を要求し，さらに, 135/tcp へポートスキャンを行うことがわかった．なお，この 135/tcp へのスキャンに対しては疑似インターネット内の nepenthes が応答し, DCOM 脆弱性を突くシェルコードが送信されることがわかった．

4 考察

前章の評価実験から，マルウェアに対するサーバ応答の時間的変化を提案手法により観測可能であり，また，収集したサーバ応答をマルウェアに対して送信することで，サーバ応答に対応したマルウェアの挙動が解析できるケースが確認できた．今回の実験では，疑似クライアントによるデータ収集後，検体解析を 1 度だけ行ったが，データ収集と検体解析を継続的に行い，疑似クライアントの更新を行うことで，より多くの情報が収集できると考える．例えばマルウェアがダウンロードした実行ファイルが実行され，別のファイルをダウンロードするといった，多段階の挙動にも対応が可能となる．

提案方式の課題としては，まず疑似クライアントによって模擬を行う通信から危険性の高い通信を確実に除去する方法について更なる検討が必要である．また，提案方式は，マルウェアからサーバへの通信が反復可能であるという暗黙の前提を置いているため，乱数を用いたチャレンジレスポンスによるクライアント認証などには対応できない．また，アクセスの特徴から疑似クライアントが攻撃者に判別される可能性も考慮する必要がある．例えば，疑似クライアントが用いる IP アドレスが限定されている場合，サーバへのアクセス頻度により疑似クライアントが検知される可能性がある．そのため，疑似クライアントの用いる IP アドレスを定期的に変更する，もしくはプロキシを介してアクセスを行う等の対策を検討する必要がある．

5 まとめ

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の時間的変化を観測し，収集された応答を解析環境内にフィードバックすることでサーバの挙動の変化に応じたマルウェアの挙動を観測可能とするマルウェア動的解析システムを提案した．今後は，疑似クライアント生成時に，より適切に通信を抽出する方法，検体数を増やしての評価実験，既存のシステムとの比較を行いたい．

謝辞 本研究に関して有益なご意見を頂いた(独)情報通信研究機構 井上大介氏，鈴木未央氏に感謝する．

参考文献

[1] 畑田充弘，他：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，MWS2009（2009 年 10 月）

[2] 須藤年章, 富士原圭, "仮想インターネットを用いたボットネット挙動解析システムの評価," コンピュータセキュリティシンポジウム 2006(CSS2006), 7C-3, 2006.

[3] 星澤裕二，岡田晃市郎，太刀川剛，"動的解析による BOT コマンドの自動抽出," マルウェア対策研究人材育成ワークショップ 2008 予稿集 (MWS2008),2008.

[4] U. Bayer, C. Kruegel, and E. Kirda, "TTAnalyze: A Tool for Analyzing Malware," 15th Annual Conference of the European Institute for Computer Antivirus Research, 2006.

[5] D. Inoue, K. Yoshioka, M. Eto, Y. Hoshizawa, K. Nakao, "Malware Behavior Analysis in Isolated Miniature Network for Revealing Malware's Network Activity," IEEE ICC 2008, pp. 1715-1721, 2008.

[6] S. Miwa, T. Miyachi, M. Eto, M. Yoshizumi, and Y. Shinoda, "Design and Implementation of an Isolated Sandbox with Mimetic Internet Used to Analyze Malwares," Proc. DETER Community Workshop on Cyber Security Experimentation and Test, 2007.

[7] K. Yoshioka, T. Kasama, T. Matsumoto, “Sandbox Analysis with Controlled Internet Connection for Observing Temporal Changes of Malware Behavior,” The Fourth Joint Workshop on Information Security (JWIS 2009)

[8] 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/

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

[10] NORMAN Sandbox Information Center,

http://www.norman.com/microsites/nsic/