マルウェア動的解析オンラインサービスの脆弱性

吉岡克成　　細渕嘉彦　　織井達憲　　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

yoshioka@ynu.ac.jp, {hosobuti, orii}@mlab.jks.ynu.ac.jp, tsutomu@ynu.ac.jp

あらまし 近年，任意のユーザからの実行ファイル等の検体の提出を受け付け，解析環境(サンドボックス)内で実行し，その挙動を解析して結果をユーザに提供するといった「マルウェア動的解析オンラインサービス」が人気を集めている．しかし，解析対象の検体が動作中にインターネットにアクセスすることを許す動的解析方法をとる場合，攻撃者は解析サービスに送り込む検体を特別に設計し，検体のアクセス先ホストと連携することで，サンドボックスのIPアドレスなどの，サービスを実施するシステムに関する情報を調べることが可能なはずである．更には，このような情報により，C&Cサーバやファイルサーバといった攻撃者が管理するサーバへの，サンドボックス内の検体からのアクセスと，サンドボックス外の検体からのアクセスを区別できるため，動的解析オンラインサービスでは十分にその機能が調べられないようにマルウェアを設計することも可能となるはずである．本報告では，このような脆弱性が現在運用されている多くのマルウェア動的解析オンラインサービスに現実に存在することを示す．

Vulnerability of Malware Sandbox Analysis as an Online Service

Katsunari Yoshioka　　Yoshihiko Hosobuchi　　Tatsunori Orii　　Tsutomu Matsumoto

Yokohama National University

79-7 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan

yoshioka@ynu.ac.jp, {hosobuti, orii}@mlab.jks.ynu.ac.jp, tsutomu@ynu.ac.jp

Abstract In recent years, malware sandbox analysis as an online service, which receives online submissions of possibly malicious executables from an arbitrary user, analyzes their behavior by actually executing them in a testing environment (i.e., a sandbox), and sends analysis reports back to the user, is becoming popular. However, we point out that an IP address of an Internet-connected sandbox could be discovered by an attacker who submits a dummy sample designed to connect to a server in attacker's control. Moreover, in order to disturb or avoid the sandbox analysis, malware authors could design their malware to conceal its behavior if it uses the discovered IP address when connecting to their Command and Control (C&C) server or file server. In this paper, we show that such vulnerability actually exists in many existing online analysis services.

1　はじめに

近年, コンピュータウイルスやワーム, ボット, トロイの木馬やスパイウェアのようなマルウェアに起因するセキュリティ脅威が我々の実生活に影響を及ぼすような深刻な問題となっている．その対策として, 解析対象の検体を解析環境(サンドボックス)内で実際に実行し，その挙動を観測・分析するマルウェア動的解析の研究が広く行われている[1-5,7-10,13-17,19-23,25,26]．また，インターネット上で実行ファイル等の検体を受け付け，自動的に動的解析を行い，解析レポートを検体投稿者に提供するサービス[15-17,20-23,25,26]が広く運用されている．これらのサービスは Windows の実行ファイルを解析対象としているが，JavaScript[25]や Flash [25], DLL[20], PDF[20,25], Web サイト[15,19,25]の解析を行うサービスも存在する．論文[3]ではオンラインサービスの 1 つである Anubis が 2 年未満の期間で 90 万検体以上(MD5 ハッシュ値によるカウント)の投稿を受けたと報告されており，このようなサービスの人気を示している．本論文では，このようなマルウェア動的解析オンラインサービスを MSaaS (Malware Sandbox analysis as an online Service)と呼ぶ．近年のマルウェアは，C&C メッセージの受信，アップデートファイルの受信，インターネットへの接続確認といった様々な理由のために外部のホストと通信することがある．そのため，サンドボックス内で実行される検体のインターネット接続を許可することが多い．このような方式を以降ではインターネット接続型の解析と呼ぶ．インターネット接続型の解析では，サンドボックスを攻撃者が管理するサーバ群と接続するため，サンドボックスの存在を攻撃者に検知されないようにしなければならない．特に MSaaS では，攻撃者は通常のユーザとして様々な検体をシステムに投稿し，システムの構成を調査することができるため，注意が必要である．

本論文では，インターネット接続型解析サービスを提供する MSaaS が本質的に内包する, 重大な脆弱性を指摘する．すなわち，解析サービスに送り込む検体を特別に設計し，検体のアクセス先ホストと連携することで, サンドボックスの IP アドレスを特定する，IP アドレス特定攻撃が可能であ

ることを示す．攻撃者は，サンドボックスの IP アドレスを特定することで，サンドボックス内の検体からのアクセスとサンドボックス外の検体からのアクセスが区別し，サーバからの応答を変えることで，サンドボックスによる解析を回避できる．現在運用されている 9 つのシステムについて，上記の IP アドレス特定攻撃に関する脆弱性が存在するか検証するケーススタディを行った結果，6 つのサービスにおいてインターネット接続型解析が行われており，いずれのサービスも IP アドレス特定攻撃に対して脆弱であることがわかった．さらに，解析サービス間での検体の授受や，解析システムから外部サーバへの定期的なアクセス(追跡調査と考えられる)などの解析活動も攻撃者によって検知されることがわかった．

本稿の構成は次のとおりである．2 章で関連研究について，3 章で MSaaS と IP アドレス特定攻撃の基本概念について，4 章で実運用されている MSaaS に関するケーススタディについて，5 章で IP アドレス特定攻撃の対策について説明し，6 章でまとめを行う．

2 関連研究

マルウェア動的解析の手法は，サンドボックスのインターネットへの接続の観点から隔離型のサンドボックスとインターネット接続型のサンドボックスに分類される．

前者の例として，ネットワーク環境を模擬した仮想インターネットへの接続を行い，実インターネットへの接続を許可しない Norman Sandbox[21]がある．Norman Sandbox は多くのネットワークサービス(HTTP，FTP，SMTP，DNS，IRC，P2P など)を模擬した隔離環境において解析を行う．先行研究[7,8,10,14]でも仮想ネットワークサービスを用いた隔離型の解析が行われている．これらの手法の問題は，マルウェアが行う多種多様な通信に対して，実インターネットを完全に模擬することが困難である点である．その上，攻撃者はマルウェアと自らが制御するサーバ間の通信に任意の独自プロトコル[11]を利用することができるため，ネットワークサービスの模擬がさらに困難になっている．

近年のマルウェアは，C&C メッセージの受信，アップデートファイルの受信，インターネットへの接続確認，現在時刻情報の取得といった様々な理由のために外部のホストと通信することがある．そのため，後者の手法のように実インターネットへの接続を許可したサンドボックスを利用することも多い．その例として CWSandbox[13,17,22]，Anubis[15]，Joebox[20]がある．これらのサンドボックスでは，それぞれ独自のポリシーに基づいて検体のインターネット接続を許可していると思われるが，その詳細は明らかにされていない．これらの手法の問題点は，攻撃者が管理する C&C サーバやファイルサーバに対して，同一の IP アドレスを用いて頻繁にアクセスすると，解析環境を攻撃者に検知され，解析回避や解析妨害のために，C&C サーバやファイルサーバのレスポンスを変更されてしまう可能性がある点である．

3 基本概念

3.1 インターネット接続型 MSaaS のモデル

インターネット接続型 MSaaS のモデルを図 1 に示す．MSaaS の解析対象として，実行可能コードなどのファイルと Web サイトがある．投稿者は解析対象のファイルまたは Web サイトの URL をシステムに投稿し，解析を依頼する．受付は解析対象ファイルや URL を受理するための，公開されたインターフェイスであり，典型的には Web サイトとして実現される．解析対象がファイルの場合，投稿された検体はサンドボックス内で実行・解析され，解析レポートが投稿者に提供される．この際，サンドボックスはインターネットと接続されており，検体は外部ホストとの通信が可能である．一方，対象が Web サイトの場合，受付が解析対象の URL を受理すると，サンドボックス内の Web ブラウザは当該 Web サイトにアクセスする．Web サイトが他のサイトのコンテンツを参照している場合，それらのサイトにもアクセスし，コンテンツを入手する．

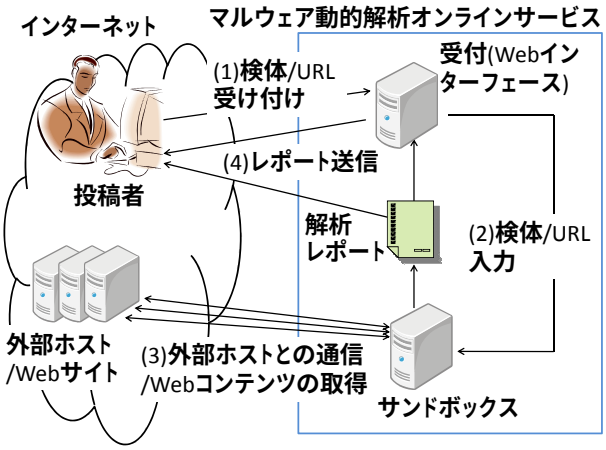

図 1　インターネット接続型 MSaaS のモデル

3.2 MSaaS の評価項目

マルウェア動的解析システムの評価項目として，論文[14]で以下の項目が挙げられている．MSaaS はマルウェア動的解析システムの一種であり，MSaaS にも当てはまると言える．

- Observability

動的解析によりマルウェアの様々な挙動をどの程度観測できるかの指標を Observability と呼ぶ．近年のマルウェアにはデバッガや仮想化システムを検知することで解析の回避を試みるものが存在する[6]．これらのマルウェアは解析環境を検知して実行停止や異なる挙動を示すため，マルウェア

の本来の挙動の観測が難しくなっている．本研究で取り上げる IP アドレス特定攻撃も解析環境を検知する攻撃であり，Observability の低下をもたらすと考えられる．

- 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かどうかの指標を Containment と呼ぶ．サンドボックスを外部のネットワークから完全に隔離する方法は高い Containment を有するが，マルウェアの外部ホストとのやり取りを完全に再現することは難しいため，Observability が低下することが一般的であり，トレードオフの関係にある．

- Efficiency

マルウェアの挙動を安定的かつ効率的に観測できるかどうかの指標をEfficiencyと呼ぶ．解析に要する時間や解析自動化の可能性もEfficiencyの重要な評価指標と考えられる．

3.3 IP アドレス特定攻撃

本節では，MSaaS への IP アドレス特定攻撃の概念を説明する．図 2 に攻撃の流れを示す．まず，攻撃者は自らが制御可能なサーバ(これを情報収集サーバと呼ぶ)への接続を行うダミー検体を作成する．ダミー検体は情報収集サーバに接続する際にユニークな ID を用いて通信するように設計する．ID はマルウェアの挙動として典型的な，ファイルリクエストや IRC のコマンドの一部として埋め込むことができる．例えば，HTTP GET のファイル名や IRC の USER， JOIN，NICK におけるユーザ名，ニックネーム，チャネル名などを ID として用いることができる．

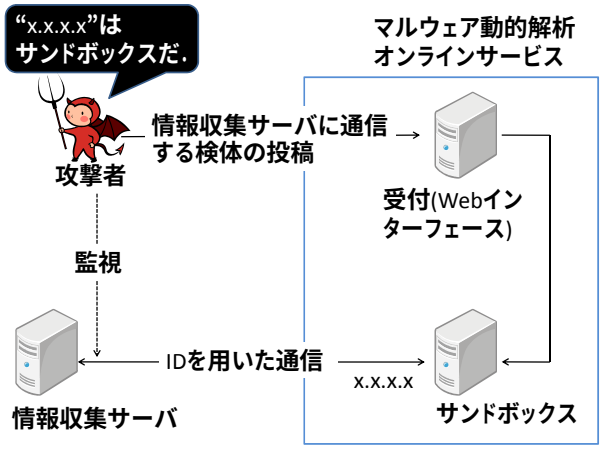

図 2　IP アドレス特定攻撃のモデル

ダミー検体を準備した後，攻撃者は解析システムにダミー検体を投稿すると共に情報収集サーバへの接続を監視する．投稿されたダミー検体はサンドボックス内で実行され，IDを用いて情報収集サーバに接続を行う．当該IDを用いて接続を行うクライアントは，先に投稿したダミー検体のみであるため，攻撃者はこのクライアントのIPアドレスをサンドボックスのIPアドレスとみなす．このように特定されたIPアドレスはブラックリストとして攻撃者のコミュニティ内で共有され，悪用される恐れがある．具体的には，当該アドレスを用いたアクセスに対して，攻撃者は自らが管理するC&Cサーバやファイルサーバの接続を拒否する等の方法で解析を妨害する可能性がある．

4 ケーススタディ

4.1 準備

ケーススタディは，実行可能コードなどのファイルの解析を行う MSaaS を提供するシステム 7 種類，Web サイトの解析を行うシステム 2 種類に関して行った．

4.1.1 ファイルの解析を行う MSaaS の検証

今回のケーススタディでは，以下のような挙動を示す単純なダミー検体を用いた．

(1) ダミー検体は最初に情報収集サーバのドメイン名の名前解決を行い，サーバに接続する．名前解決が出来ない場合は停止する．
(2) TCP セッションが確立されると，HTTP GET リクエストを送信する．TCP セッションが一定時間内に確立されない場合は停止する．また，リクエストするファイル名を ID として使用する．
(3) サーバからファイルを受信したら，当該ファイルがキーワード文字列を含むかどうか検査する．ファイルが一定時間内に受信できない場合は停止する．
(4) 受信したファイル内にキーワード文字列が見つかったら，隠された挙動(別ファイルの作成と，別のサーバへの通信)を実行する．見つからない場合は停止する．

上記のダミー検体は，ユニークな ID を用いて情報収集サーバにアクセスし，サンドボックスの IP アドレスを暴露する機能と，受信したファイル内のキーワード文字列の有無に応じて，挙動を変化させる機能を持つ．

次に情報収集サーバについて説明する．情報収集サーバはターゲットとなる MSaaS のサンドボックスが使用する IP アドレスのリスト(IP アドレスブラックリスト)と，投稿されるダミー検体群が用いる ID のリスト(有効 ID リスト)を保持し，以下のような処理を行うように設計した．

(1) クライアントからの接続要求を待ち受ける．
(2) HTTP GET リクエストを受信したら，要求ファイル名(ID)が有効 ID リストに含まれるか確認する．
(3) 有効な ID であれば，そのクライアントが使用している IP アドレスを IP アドレスブラックリストに追加する．さらに，設定に応じてキーワード文字列が含まれたファイルをクライアントへ送信する．

情報収集サーバは，HTTP GET リクエストを受信し，ID の有効性確認を行った後に IP アドレスブラックリストの更新を行う機能と，キーワードを含むファイルをトリガーとしてダミー検体に送ることで，ダミー検体に対して隠された挙動の実行を指示する機能を持つ．今回のケーススタディでは，有効な ID を用いた HTTP GET リクエストのうち，送信元 IP アドレスが実験期間中に初めて使用された場合のみ，トリガーを送信するように設定した．ダミー検体と情報収集サーバの連携の流れを図 3 に示す．

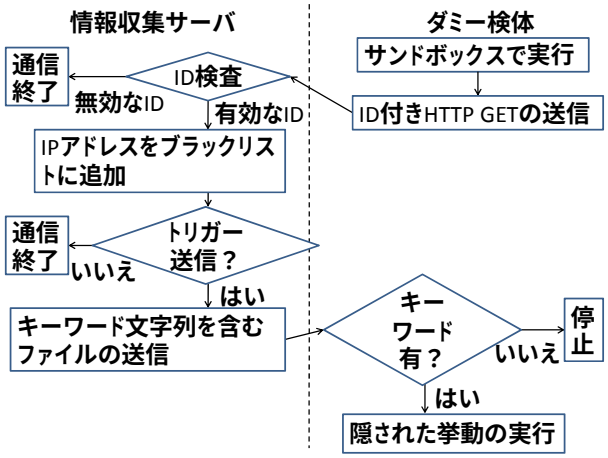

図 3　ダミー検体と情報収集サーバの連携

4.1.2　Web サイトの解析を行う MSaaS の検証

Web サイトの解析を行う MSaaS の検証は以下のように行った．まず，4.1.1 節で述べた情報収集サーバとは別に新たな情報収集サーバ(例:www.fake.com)を用意した．次に，HTML ファイル名(例:xyz123.html)を ID として，URL (例:http://www.fake.com/xyz123.html)を生成し，これを解析対象 URL として解析システムに解析依頼を行った．一方，情報収集サーバでは 4.1.1 節と同様に IP アドレスブラックリストの保持・更新と有効 ID リストの保持を行うと共に，HTTP GET リクエストに対して，2 種類の HTML ファイル($HTML_{Normal}$ と $HTML_{Malicious}$)のいずれかを返信するように設計した．

今回のケーススタディでは，通常の HTTP GET リクエストに対しては，$HTML_{Malicious}$ ファイルを返信し，送信元 IP アドレスが IP アドレスブラックリストに含まれる場合に限り，$HTML_{Normal}$ を返信するように設定した．なお，実際の IP アドレス特定攻撃では $HTML_{Malicious}$ は不正な Web コンテンツ，$HTML_{Normal}$ は解析システムを欺くための無害なダミーコンテンツであることを想定しているが，今回のケーススタディでは，いずれも無害な Web コンテンツを用いた．

4.2　方法

MSaaS を提供する実運用中の 9 種類のシステムに対し，7 日間掛けてケーススタディを行った．

ここでシステム 1 から 7 は実行可能コードなどのファイルを解析対象とするシステムであり，システム 8，9 は Web サイトを解析対象とするシステムである．それぞれのシステムに対するケーススタディの方法を以下に示す．

- **ファイル対象システムの検証方法**

(1) ユニークな ID を持つ 245 個のダミー検体を用意した．さらに情報収集サーバの IP アドレスブラックリストを空に設定し，上記の 245 のダミー検体が用いる ID を全て有効 ID リストに登録した．
(2) 245 個の検体を，35 個ずつの検体からなる 7 グループに分割し，1 つのシステムに 1 つのグループを割り当てた．
(3) 各システムに割り当てられた 35 個のダミー検体をさらに 7 個のサブグループに分割し，毎日 1 つのサブグループに含まれる検体を当該システムに投稿した．すなわち，各システムに対して毎日 5 検体ずつ 1 週間に渡ってダミー検体を投稿した．
(4) 情報収集サーバは，有効な ID を用いた HTTP GET リクエストのうち，送信元 IP アドレスが実験期間中に初めて使用された場合に限りキーワード文字列を含むファイルをトリガーとして返信するように設定した．
(5) 各解析システムから解析レポートを収集した．

- **Web サイト対象システムの検証方法**

(1) ユニークな ID を含む 70 個の URL を用意した．さらに，情報収集サーバの IP アドレスブラックリストを空に設定し，上記の 70 の URL が用いる ID を全て有効 ID リストに登録した．
(2) 70 個の URL を，35 個ずつの URL からなる 2 グループに分割し，1 つのシステムに 1 つのグループを割り当てた．
(3) 各システムに割り当てられた 35 個の URL をさらに 7 個のサブグループに分割し，毎日 1 つのサブグループに含まれる URL の解析を当該システムに依頼した．すなわち，各システムに対して毎日 5 つの URL の依頼を，1 週間に渡って依頼した．
(4) 情報収集サーバは，有効な ID を用いた HTTP GET リクエストのうち，送信元 IP アドレスが実験期間中に初めて使用された場合に限り，$HTML_{Malicious}$ を返信し，それ以外の GET リクエストに対しては $HTML_{Normal}$ を返信するように設定した．
(5) 各解析システムから解析レポートを収集した．

4.3　結果

表 1 に検証結果を示す．9 つのシステムの内，6 つがインターネット接続型の解析手法を採用しており，いずれも IP アドレス特定攻撃に対して脆弱であることがわかった．すなわち，これらのシス

テムのサンドボックスが用いる IP アドレスを容易に特定できると共に，情報収集サーバの挙動の制御により，攻撃者の意図通りに不正な挙動や不正な Web コンテンツを隠蔽可能であることがわかった．残りの 3 システムは全く情報収集サーバにアクセスがなかったことから，インターネット接続型の解析を採用していないものと考えられる．この場合は，そもそも解析対象が外部サーバにアクセスできないため，隠された挙動や不正な Web コンテンツの観測は出来ず，十分な解析結果は期待できない．

表 1　IP アドレス特定攻撃への脆弱性検証結果

システム	解析レポート	情報通信サーバへの通信	IP アドレスの発見	挙動の隠蔽	解析対象
1	有	有	可能	可能	File
2	有	無	適用せず	可能	File
3	有	有	可能	可能	File
4	有	無	適用せず	可能	File
5	有	有	可能	可能	File
6	有	無	適用せず	可能	File
7	有	有	可能	可能	File
8	有	有	可能	可能	Web
9	有	有	可能	可能	Web

表 2　検証結果の詳細

システム	解析レポート記載		情報収集サーバに実際に接続	特定 IP アドレス数	隠された挙動の報告
	DNS クエリ	HTTP GET			
1	35	35	35	1	1
2	35	35	0	0	0
3	35	35	35	2	2
4	35	0	0	0	0
5	35	35	35	10	10
6	35	35	0	0	0
7	35	35	35	1	1
8	35	35	35	12	12
9	35	35	35	1	1

表 3　興味深い情報収集サーバへの通信

ケース	使用 ID	IP アドレス数	特徴	推測
1	システム 3	1	ダミー検体とは異なるフォーマットの HTTP GET	解析システムによる自動監視
2	システム 2	1	ダミー検体とは異なるフォーマットの HTTP GET	解析システムによる自動監視
3	システム 2	1	1.5 時間毎の WGET による HTTP GET	解析システムによる自動監視
4	システム 5	11	ダミー検体/それ以外による HTTP GET	解析システムによる自動/手動監視
5	システム 5	1	システム 7 の IP アドレスによる HTTP GET	システム間の検体共有
6	システム 5	2	ケース 2,3 の IP アドレスによる HTTPGET	システム間の検体共有
7	システム 3	1	ケース 3 の IP アドレスによる HTTPGET	システム間の検体共有
8	なし	6	TCP SYN 受信と TCP セッション確立	Web サーバへの攻撃・偵察
9	なし	3	ドキュメントルートへの HTTP GET	Web サーバへの攻撃・偵察
10	なし	1	ドキュメントルートへの HTTP HEAD	Web サーバへの攻撃・偵察

表 2 に検証結果の詳細を示す．表 2 の通り，全てのシステムにおいて，情報収集サーバに接続するための最初の挙動である DNS クエリが観測され, 解析レポートで報告されていることがわかる．このうち, システム 4 を除く 8 つのシステムでは，HTTP GET のリクエストが観測されている．システム 2 と 6 は実際に情報収集サーバへの接続を行っていないことから，これらのシステムでは，隔離環境による解析を採用しており，解析環境内に模擬 Web サーバを用意していると思われる．

検証期間中に使われた IP アドレス数に注目すると，システム 1，7，9 では，1 つのアドレスを継続して使っており，解析環境であることが容易に露見する恐れがある．システム 3 では，最初の 6 日間は 1 つのアドレスを使っていたが，最終日だけ 2 のアドレスを使っていた．システム 5，8 ではそれぞれ 10 個，12 個のアドレスを使っていたが，これらのアドレスは，全て同じネットワークセグメントに属しており，当該セグメント全体がサンドボックス解析に利用されていると攻撃者が推測することは容易と言える．

4.4　情報収集サーバへの興味深い通信

今回のケーススタディでは，投稿したダミー検体からのアクセスに加えて，情報収集サーバへの興味深い通信が多数観察できた. それらを 10 のケースにわけ，表 3 に示す．ケース 1 から 7 は，有効な ID による HTTP GET リクエストであり，投稿した検体との関連があると言えるが，ケース 8 から 10 は投稿した検体との関連性は明らかでない．それぞれのケースについて以下に説明する．

- **情報収集サーバの自動継続監視**

ケース 1，2，3 では，システムにダミー検体を投稿し動的解析が終了した一定時間後に，情報収集サーバへの HTTP GET リクエストを再び観測した．これらのリクエストのフォーマットは，ダミー検体のそれとは異なっていることから，何らかの自動化された継続監視用ツールによるアクセスだと考えられる．特に，ケース 3 では WGET[18] によるリクエストが，1.5 時間毎に 3 日間続いた．これらの通信は 1 つの IP アドレスで行われており，攻撃者に検知される可能性は高い．

- **システムによる自動/手動監視**

ケース 4 では, システム 5 に投稿した検体の ID による HTTP GET リクエストを検証期間内に複数回受信した．これらの通信には 11 個の異なる IP アドレスが使われており，リクエストのフォーマットから，ダミー検体による通信とそれ以外による通信があったことがわかった．

- **システム間の検体授受**

ケース 5 から 7 では，あるシステムに投稿したダミー検体の使用 ID を含む HTTP GET リクエストが, 別のシステムの IP アドレスを用いて送信されたことを確認した．これは，前者のシステムから後者のシステムにダミー検体が提供されたためと考えられる．ケース 5 では，システム 5 から 7 へ，ケース 6 ではシステム 5 から 2 へ，ケース 7

ではシステム 3 から 2 へ提供されていることがわかり，解析システム間の検体授受の流れの一部が攻撃者により観測可能であることがわかった．

- **その他のアクセス**

ケース 8 から 10 は通信に ID を含んでおらず，投稿したダミー検体に関係あるかどうかは明らかではないが，Web サーバの脆弱性を狙った攻撃およびスキャンと考えるのが妥当と言える．

5 IP アドレス特定攻撃の対策

5 章では，IP アドレス特定攻撃の対策について議論する．IP アドレス特定攻撃への自明な対策は，解析に毎回異なる IP アドレスを使用することである．しかし，一般に，解析システムが使用できる IP アドレス数は，解析対象のマルウェア検体数に比べて少ないため，この対策は現実的でない．

- **投稿者をプロキシとして利用する**

この問題の解決方法の 1 つは投稿者から協力を得ることである．すなわち，図 4 に示すように，解析のために投稿者の IP アドレスを一時的にプロキシとして利用し，受付を介して多段プロキシ化することである．この場合，攻撃者は投稿者の IP アドレスしかわからないため，解析システムの IP アドレスを隠蔽することができる．このアプローチの問題点は，マルウェアが生成する潜在的に悪意のある通信が，投稿者のマシンを経由することである．もう 1 つの問題点は，投稿者は自らのマシンにプロキシツールをインストールし，解析中にツールを実行しなければならないことである．また，攻撃者に多段プロキシを検知されないようにする必要がある．

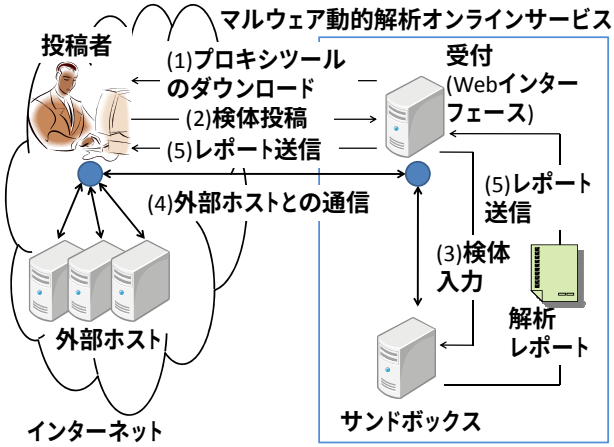

図 4　投稿者プロキシ MSaaS のモデル

- **投稿者の行動の分析**

もうひとつの対策は，投稿者の行動の分析である．例えば，投稿者(の IP アドレス)を基にプロファイリングを行い，通常のユーザと攻撃者のシステム利用における行動を区別する．

- **複数の解析サービスの併用**

多くのセキュリティサービスと同様に，最も重要な解決法は，サービスの利用者自体がシステムの特徴と限界を理解することである．MSaaS とオンラインファイルスキャン[24]などのサービスとを併用することで，個々のサービスの限界を補完することが重要といえる．

6 まとめ

本論文では，インターネット接続型 MSaaS が本質的に内包する問題である，IP アドレス特定攻撃への脆弱性を指摘すると共に実運用されている多くのシステムにおいて，実際にこのような問題が存在することを示した．今後の課題は効果的な対策案に関する更なる考察と対策の実装評価である．

参考文献

[1] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, “Automated Classification and Analysis of Internet Malware,” Proc. of Recent Advances in Intrusion Detection, RAID07, LNCS Vol. 4637, pp. 178-197, 2007.
[2] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda. “Scalable, Behavior-Based Malware Clustering.” In Symposium on Network and Distributed System Security (NDSS), 2009.
[3] U. Bayer, I. Habibi, D. Balzarotti, E. Kirda, and C. Kruegel, "A View on Current Malware Behaviors," Proc. 2nd Usenix Workshop on Large-Scale Exploits and Emergent Threats, LEET'09, 2009.
[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 (EICAR), 2006.
[5] M. Becher, and F. Freiling, "Towards Dynamic Malware Analysis to Increase Mobile Device Security," Proc. of SICHERHEIT 2008.
[6] X. Chen, J. Andersen, Z. M. Mao, M. Bailey, J. Nazario, “Towards an Understanding of Anti-virtualization and Anti-debugging Behavior in Modern Malware,” Proc. International Conference on Dependable Systems and Networks, pp 177 – 186, 2008.
[7] D. Inoue, K. Yoshioka, M. Eto, Y. Hoshizawa, K. Nakao, "Malware Behavior Analysis in Isolated Miniature Network for Revealing Malware's Network Activity," IEEE International Conference on Communications (ICC 2008), pp. 1715-1721, 2008.
[8] D. Inoue, K. Yoshioka, M. Eto, Y. Hoshizawa, and K. Nalao, "Automated Malware Analysis System and its Sandbox for Revealing Malware's Internal and External Activities," IEICE Trans. Vol. E92D, No. 5, 2009.
[9] E. Kirda, C. Kruegel, G. Banks, G. Vigna, and R. Kemmerer. "Behavior-based Spyware Detection." Usenix Sec, 2006.
[10] 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.
[11] P. Porras, H. Saidi, and V. Yegneswaran. “A Foray into Conficker’s Logic and Rendezvous Points.” Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2009.
[12] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A Multifaceted Approach to Understanding the Botnet Phenomenon,” ACM SIGCOMM Conf. on Internet Measurement, pp 41 - 52, 2006.
[13] 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.
[14] 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, 2009.
[15] Anubis, http://analysis.seclab.tuwien.ac.at/.
[16] Comodo Instant Malware Analysis, http://camas.comodo.com/cgi-bin/submit
[17] CWSandbox, http://www.cwsandbox.org/
[18] GNU WGET, http://www.gnu.org/software/wget/
[19] gred, https://www.gred.jp/?tab=goleo
[20] Joebox, http://www.joebox.org/.
[21] Norman Sandbox,http://www.norman.com/technology/norman_sandbox/
[22] Sunbelt CWSandbox, Malware Research Labs, http://www.sunbeltsecurity.com/
[23] Threat Experts, http://www.threatexpert.com/
[24] Virustotal, http://www.virustotal.com/
[25] Wepawet, http://wepawet.iseclab.org/
[26] Zero Wine, http://zerowine.sourceforge.net/