情報処理学会論文誌 Vol. 52 No. 9 2761–2774 (Sep. 2011)

推薦論文

# 公開型マルウェア動的解析システムに対するデコイ挿入攻撃の脅威

笠間貴弘[†1] 織井達憲[†1]
吉岡克成[†1] 松本勉[†1]

近年，任意のユーザから実行ファイルなど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「公開型マルウェア動的解析システム」が人気を集めている．我々はこれまで，特別に設計された検体（デコイ）をシステムに提出することで，サンドボックスの情報を暴露させ，その情報を基にサンドボックスの検知を行う攻撃手法として「デコイ挿入攻撃」を提案し，実証実験により，サンドボックスの IP アドレスを用いた検知が実運用中の 15 個のシステムに対して有効であることを示している．しかし，IP アドレス以外の情報を用いた検知については未検証だった．当該脆弱性を正確に把握し適切な対策を導出するため，本稿では，まず，IP アドレスを含む 16 種類のサンドボックス情報に着目し，これらの情報が，取得安定性や個別性といった，サンドボックス検知に有効な性質を有しているかを実証実験により評価する．実験の結果，Windows プロダクトキー，MAC アドレス，OS インストール日時といったサンドボックス情報は，検知対策を行っていると思われる特定の例外を除いて，検知に利用できることが分かった．さらに，ネットワークを介さずに解析レポート経由でサンドボックス情報を暴露する方法も有効であり，我々のこれまでの検討では攻撃対象となりえなかった，隔離型サンドボックスも攻撃対象となりうることが確認された．このことから，公開型マルウェア動的解析システムにおいては，IP アドレス以外のサンドボックス情報による検知への対策や，解析レポート経由による暴露への対策などを含めた，総合的なデコイ挿入攻撃への対策が必要であることが分かった．

本論文の内容は 2010 年 10 月のコンピュータセキュリティシンポジウム 2010 にて報告され，同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

# On the Power of Decoy Injection which Threatens Public Malware Sandbox Analysis Systems

Takahiro Kasama,[†1] Tatsunori Orii,[†1]
Katsunari Yoshioka[†1] and Tsutomu Matsumoto[†1]

Recently, the use of public Malware Sandbox Analysis Systems (public MSASs) which receive online submissions of possibly malicious files or URLs from an arbitrary user, analyze their behavior by executing or visiting them by a testing environment (i.e., a sandbox), and send analysis reports back to the user, has increased in popularity. In previous study, we have pointed out a vulnerability of public MSASs against decoy injection attack, in which an attacker detects the sandbox based on its IP address which can be obtained by submitting a decoy sample designed for this purpose. However, we did not further investigate the possibility of detection using sandbox information other than its IP address. In this paper, in order to better understand the vulnerability and develop an effective countermeasure, we evaluate 16 different kinds of characteristics in the sandbox in terms of their accessibility and uniqueness for sandbox detection. As a result of experiments with real public MSASs in operation, we found that characteristic information such as Windows' product key, MAC address, and system install time can be utilized for sandbox detection, except for particular systems which appeared to have deployed a countermeasure. Moreover, besides network-based disclosure, we show that such characteristic information of the sandbox can be disclosed via an analysis report provided to the user, which means that the decoy injection attack can be performed against the sandbox isolated from the real Internet. Thus, our study confirmed the broad applicability of the decoy injection attack and also necessity of comprehensive countermeasures.

## 1. はじめに

近年，ワームやボット，トロイの木馬，スパイウェアといった悪意のあるソフトウェア（マルウェア）によるセキュリティ被害が深刻な社会問題となっている．これに対し，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，挙動を明らかにするマルウェア動的解析の研究が広く行われている[1)–4),9)–12),17),18),20),23),27),36)]．しかしそのような解

†1 横浜国立大学
Yokohama National University

析システムを一般のユーザが独自に構築し，解析を行うことは難しい．そこでインターネット上で実行ファイルなどの検体を受け付け，自動的に動的解析を行い，解析結果を解析レポートとして検体投稿者に提供する，「公開型マルウェア動的解析システム」が多くのセキュリティベンダや研究機関によって運用されている[22)–28),32)–36),38),43),45)–47),49)–51)]．これらのシステムの多くは Windows の実行ファイルを解析対象としているが，JavaScript[50)] や Flash[50)]，DLL[33)]，PDF[33),50)]，Web サイト[22),23),28),32),34),35),38),46),49)–51)] の解析を行うシステムも存在する．本稿では，このようなシステムを Public MSAS（Public Malware Sandbox Analysis Systems）と呼ぶこととする．我々は先行研究[19),21)] において，攻撃者が特別に設計された検体（これをデコイと呼ぶこととする）をシステムに提出することで，サンドボックスの情報を暴露させ，その情報を基にサンドボックスの検知を行う攻撃手法として「デコイ挿入攻撃」を提案し，実運用中の 15 個のシステムに対する実証実験により，サンドボックスの IP アドレスを用いた検知が有効であることを示している．しかし，IP アドレス以外の情報の検知への適用可能性については検証を行っていなかった．

そこで本稿では，まず，検知を行ううえで有効なサンドボックス情報の性質として，取得安定性，個別性，ステルス性を定義し，実際に前述のシステムから 16 種類のサンドボックス情報の収集を試みることで，取得安定性と個別性をそれぞれ評価した．実験の結果，Windows プロダクトキー，MAC アドレス，OS インストール日時といったサンドボックス情報は，個別に検知対策を行っていると思われるいくつかの例外を除いて，高い取得安定性と個別性を有しており，検知に利用される可能性があることが分かった．さらに，ネットワークを介さずに解析レポート経由でサンドボックス情報を暴露する方法の有効性も実証実験により確認した．このことから，公開型マルウェア動的解析システムにおいては，IP アドレス以外のサンドボックス情報による検知への対策や，解析レポート経由による暴露への対策などを含めた，総合的なデコイ挿入攻撃への対策が必要であることが分かった．

まず 2 章で関連研究について述べた後，3 章で Public MSAS の分類を行い，Public MSAS の評価項目を示す．4 章で，デコイ挿入攻撃を説明し，5 章で実際に運用されている主要な Public MSAS 15 個に対して行った検証実験について説明する．6 章で考察を行い，7 章でまとめとする．

## 2. 関 連 研 究

マルウェア動的解析の既存研究はサンドボックスとインターネットの接続性の観点から大きく 2 つに分けて考えることができる．1 つは接続を許可しない隔離型，もう 1 つは接続を許可するインターネット接続型である．

前者の例としては，Norman Sandbox[36)] があげられる．Norman Sandbox では解析環境内に疑似的なインターネット環境を用意し，その中で多種多様なネットワークサービス（HTTP，DNS，SMTP，IRC など）を模擬することで，マルウェアの挙動を観測する．その他の隔離型の解析手法[9),10),12),18)] も同様に解析環境内に疑似的なインターネット環境を構築し，解析を行っている．しかし，マルウェアは C&C（Command and Control）サーバとの通信において，認証などの技術を用いたり，独自プロトコルを使用したりするなど多種多様な実装が可能なため[13),15)]，その通信のすべてを疑似インターネットで模擬することは困難である．

一方，後者の例としては Anubis[2),3),23)] があげられる．Anubis では，サンドボックスとインターネット上のホストとの通信を許可するが，インターネットへ向かう外向けの通信に関してはフィルタリングを行うことで危険性の高い通信がサンドボックス外に流出することを防いでいる．同様にインターネット接続型の解析手法の多く[1),4),17),20),27),42),43)] でも独自のポリシに基づいて通信のフィルタリングが行われているが，その詳細については明らかにされていない．これらの手法においては，リアルタイムでのフィルタリングが必要となるが，基本的に未知のマルウェアを解析対象としているため，その攻撃通信のすべてを正確にフィルタリングすることは困難である．

このように多くの動的解析手法が研究される一方で，マルウェア作成者は動的解析を回避する機能をマルウェアに搭載するようになった．たとえば，仮想化システム検知[6),8),14),41),52)] や，耐デバッグ機能[6),8)] を利用することで解析を回避したり，CPU 命令の実行時間の違いによって解析システムの検知を行ったりする[8),11),14)]．また，インターネット接続型の解析システムがサンドボックス外への攻撃をフィルタリングすることに注目して，攻撃者自身の用意したホストに対して攻撃を行い，それが成功するか否かで解析システムを検知する手法[16)] も存在する．

上記の手法は，解析システム全般において，典型的に用いられる仮想化システムやデバッガ，またはフィルタリング設定を基に解析システムを検知する手法であるが，一方で特定の解析システムを検知する手法として，当該システムのサンドボックスの特徴（Windows のプロダクトキーや特定の DLL）をチェックすることで特定の解析システムを検知する手法も存在する[3),30),31),39)]．しかし，これらの手法を適用するためには，攻撃者は事前に検知対象のサンドボックスの特徴（サンドボックス情報）を把握する必要があるが，その具体的な方法については述べられていない．また，検知に用いるサンドボックス情報に関しても，

ある特定の情報についてのみ述べられており，どのようなサンドボックス情報が検知に有効なのか網羅的な検証が行われていない．

そこで，我々は今までにサンドボックス情報を把握する段階も含めた攻撃方法として，デコイ挿入攻撃を提案している[19),21)]．デコイ挿入攻撃では，Public MSAS が任意のユーザからの検体投稿を受付ける点に注目しており，まず攻撃者は攻撃対象の Public MSAS へ特別に設計した検体（デコイ）を挿入することで当該システムのサンドボックス情報を取得する．その後，取得したサンドボックス情報を基にサンドボックスでの実行を検知し，Public MSAS での解析回避を行う．さらに，実運用されている 15 個の Public MSAS に対する実証実験によって，それらのシステムが，インターネットに接続されたサンドボックスの IP アドレスを用いた当該攻撃に対して脆弱であることを示した．

## 3. 基 本 概 念

本章では，Public MSAS の分類とモデル化を行い，Public MSAS の評価項目を示す．

### 3.1 Public MSAS の分類とモデル

本節では，まず Public MSAS を解析対象とインターネットとの接続性の観点から分類する．

まず解析対象としては，Windows の実行ファイルなどを対象とするシステムと，Web サイトを対象とするシステムに分けられる．本稿では前者を Public MSAS-F，後者を Public MSAS-W と表記する．**図 1**，**図 2** に Public MSAS-F のモデル図を示す．図 1 は隔離型のサンドボックスを，図 2 はインターネット接続型のサンドボックスを用いたシステムのモデルとなっている．

投稿者は解析対象のファイルをシステムに投稿し解析を依頼するユーザである．受付は解析対象のファイルを受理するために公開されたインタフェースであり，典型的には Web サイトとして実現される．サンドボックスは解析対象のファイルを実行し挙動を解析するための環境であり，内部ではマルウェアの挙動を把握するために各種ログ（通信ログやレジストリアクセスログ，API 呼び出し履歴など）が取得され，それらをまとめた解析レポートが投稿者に提供される．隔離型のサンドボックスの場合は，インターネットと接続する代わりに，インターネット上のサービスを模擬するための多数のサーバからなる疑似インターネットを用意することが多い．

次に，**図 3** に Public MSAS-W のモデル図を示す．Public MSAS-W では，投稿者は解析対象の URL を受付に投稿し解析を依頼する．解析対象の URL を受け取ると，サンドボッ

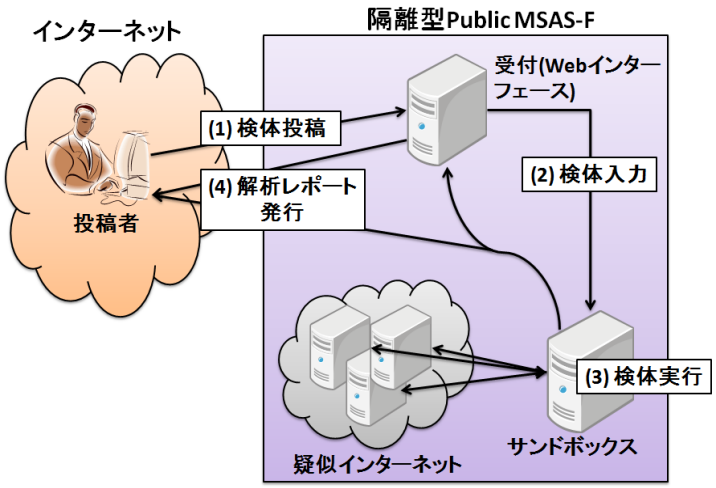


**図 1**　隔離型 Public MSAS-F のモデル図

Fig. 1　A model of public malware sandbox analysis systems for sample files (Public MSAS-F) with an isolated sandbox.

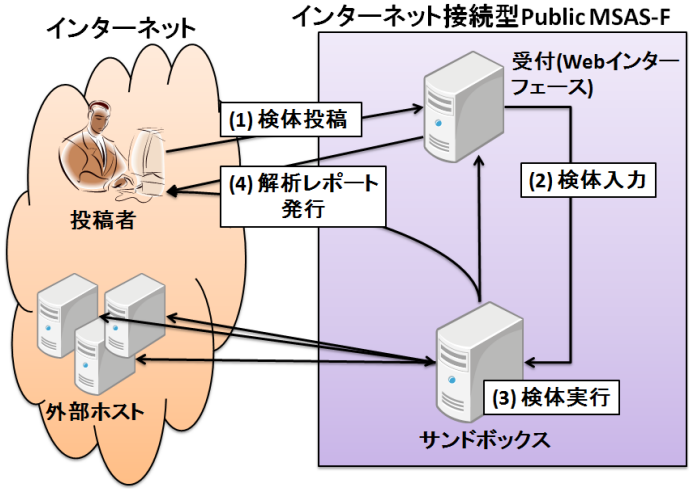


**図 2**　インターネット接続型 Public MSAS-F のモデル図

Fig. 2　A model of public malware sandbox analysis systems for sample files (Public MSAS-F) with an internet-connected sandbox.

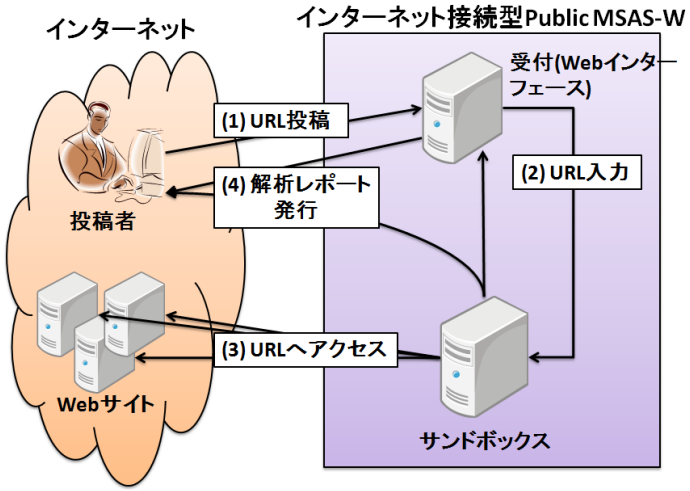


**図 3**　インターネット接続型 Public MSAS-W のモデル図
Fig. 3　A model of public malware sandbox analysis systems for web sites (Public MSAS-W).

クスは実際にその URL のコンテンツを取得し，解析を行う．解析対象の Web サイトからコンテンツ取得を行うため，Public MSAS-W は通常インターネット接続型サンドボックスを用いる．

### 3.2　Public MSAS 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 20) において以下の 3 つの項目があげられている．これらの評価項目は Public MSAS にも適用することができる．

- Observability（観測可能性）

Observability は動的解析によってマルウェアの様々な挙動を観測できる性質に関する評価項目である．

- Containment（安全性）

Containment は 2 つの評価項目からなる．1 つは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したりすることなく安全に解析を行えるかという観点での評価項目である．もう 1 つは，解析システムの検知に用いられるような，システムの重要な情報の流出を防げるかという観点での評価項目である．

- Efficiency（効率性）

Efficiency はマルウェアの挙動を安定的かつ効率的に観測できる性質に関する評価項目である．

上記の 3 つの項目はそれぞれトレードオフの関係になっている．たとえば，高い Observability を実現するためには，隔離型のサンドボックスよりもインターネット接続型のサンドボックスを用いた解析が望ましいが，インターネットに接続することでマルウェアの攻撃がサンドボックス外に流出する危険性は高まるため，Containment は低下する．

なお，本稿で述べるデコイ挿入攻撃は Public MSAS の Obsevability を低下させる攻撃の 1 種と考えることができる．

## 4.　デコイ挿入攻撃

本章では，デコイ挿入攻撃のモデルについて示す．デコイ挿入攻撃は 2 つのフェーズからなる．1 つはサンドボックス情報暴露フェーズ，もう 1 つはサンドボックス検知フェーズである．以下それぞれのフェーズについて述べる．

### 4.1　サンドボックス情報暴露フェーズ

サンドボックス情報暴露フェーズでは，攻撃者は Public MSAS に対してデコイを挿入（投稿）し，サンドボックス情報の暴露を試みる．このフェーズでの攻撃者の目的は，デコイを挿入することによって，対象の Public MSAS を特定するためのサンドボックス情報を取得し，デコイから攻撃者にその情報を伝えさせる（暴露させる）ことである．**図 4** に Public MSAS-F へのデコイ挿入のモデル図を示す．Public MSAS-F へのデコイ挿入では，攻撃者はデコイ（実行ファイル）を用意しシステムへ投稿する．投稿するデコイは，実行されると対象の Public MSAS を特定するためのサンドボックス情報を取得し，それを攻撃者に暴露する．このとき，情報の暴露方法としては，

- ネットワーク経由
- 解析レポート経由

の 2 通りが考えられる．

ネットワーク経由での暴露方法では，攻撃者はデコイを投稿する前に，暴露サーバとして自身で管理可能なサーバを用意しておく．そしてデコイをその暴露サーバに対してアクセスするように実装し，デコイと暴露サーバとの通信を通じてサンドボックス情報を暴露する．

一方，ネットワークを介さない暴露方法として，解析レポートを経由した暴露方法が考えられる．この手法では Public MSAS が投稿ユーザに提供する解析レポートに注目し，取得

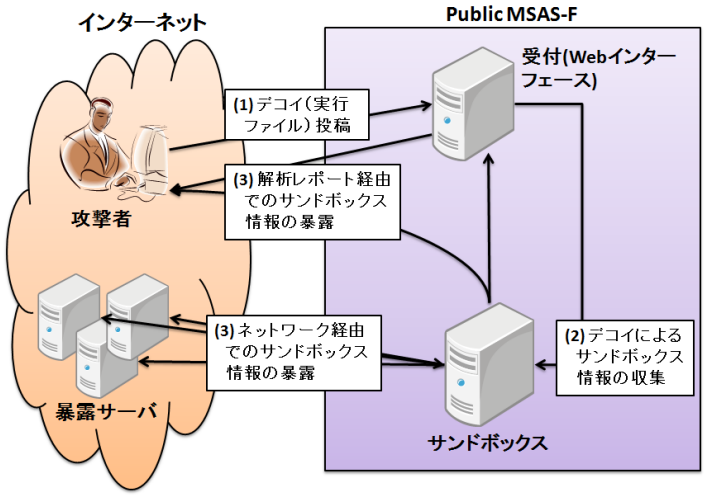


**図 4**　Public MSAS-F へのデコイ挿入のモデル図
Fig. 4　A model of decoy injection against Public MSAS-F.

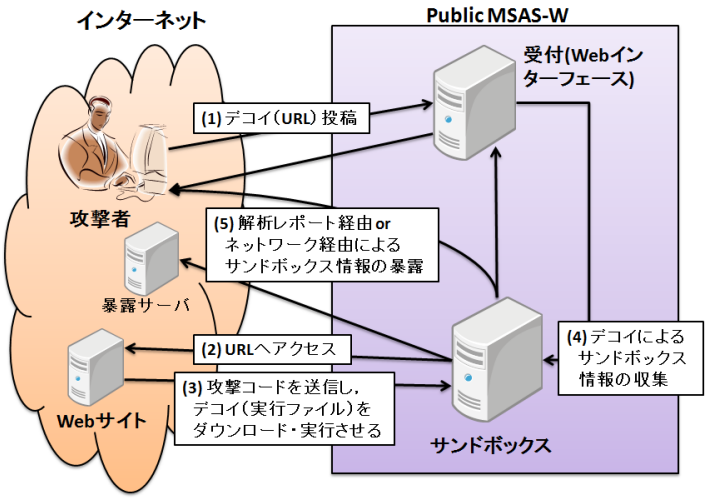


**図 5**　Public MSAS-W へのデコイ挿入のモデル図
Fig. 5　A model of decoy injection against Public MSAS-W.

したサンドボックス情報を解析レポートに埋め込むことで，暴露サーバとの通信無しに暴露を可能とする．たとえば，投稿するデコイに Windows のプロダクトキーをレジストリから取得し，同じ名前のファイルを作成する機能を付加した場合，解析レポートに検体の作成したファイル名が載ることによって，攻撃者は解析レポートを見ることでサンドボックスの Windows プロダクトキーを知ることができる．このような解析レポート経由の暴露方法の場合，隔離型の Public MSAS にも適用できるという特徴がある．

次に，**図 5** に Public MSAS-W へのデコイ挿入のモデル図を示す．Public MSAS-W へのデコイ挿入では，攻撃者はまず暴露サーバとして Web サーバを用意し，その Web サーバ上のコンテンツに対応したデコイ（URL）をシステムへ投稿する．その際，サンドボックスから暴露サーバへのアクセスを監視することで，サンドボックスの IP アドレスを特定することができる．さらに URL のコンテンツとして，Web ブラウザの脆弱性を突き，任意のファイルをダウンロード・実行させる攻撃コードを含む Web コンテンツを用意することで，デコイ（実行ファイル）をダウンロード・実行させることも可能になる．その場合，デコイ（実行ファイル）実行後の流れは Public MSAS-F の場合と同様である．そのほかにも，JavaScript などのスクリプトを送信することで Web ブラウザの情報を取得し，その情報を基にサンドボックスの Web ブラウザを識別する方法も考えられる[29]．

### 4.2　サンドボックス検知フェーズ

サンドボックス検知フェーズでは，攻撃者は暴露フェーズで取得した情報を基に，サンドボックスを検知し，Public MSAS による解析の回避を行う．その際の検知方法としては，

- ホストベースの検知
- ネットワークベースの検知

の 2 通りが考えられる．

ホストベースの検知では，マルウェア自身が検知用のサンドボックス情報のリストを保持しておき，実行時に取得したサンドボックス情報とマッチングを行う．リストと一致した場合は，挙動を切り替えることで解析の回避を行う．

一方，ネットワークベースの検知では，まず攻撃者は自身の管理するサーバに検知用のサンドボックス情報のリストを持たせておく．そのうえで，マルウェアから当該サーバへのアクセスがあった場合には，マルウェアの送信してきたサンドボックス情報とマッチングを行い，一致した場合はマルウェアへの応答を変化させることで解析の回避を行う．ホストベースの検知は隔離型のサンドボックスでも検知ができるという利点があり，ネットワークベー

スの検知は検知用のサンドボックス情報の更新が簡単だという利点がある．

### 4.3　検知に用いるサンドボックス情報

実際に検知に利用するサンドボックス情報の候補としては，Windows のプロダクトキー，MAC アドレス，IP アドレスなどの様々な情報が考えられる．そこで本節では，検知に用いるサンドボックス情報に求められる性質として，以下の 3 つの性質を示す．

- 取得安定性

取得安定性は当該サンドボックス情報が毎回安定して同じ値が取得できるという性質を示す．実際に Public MSAS による解析を回避するためには，サンドボックス情報暴露フェーズとサンドボックス検知フェーズの 2 つのフェーズにおいて，同じ値が取得できることが求められるため，検知に用いるサンドボックス情報には高い取得安定性が求められる．

- 個別性

個別性は当該サンドボックス情報が各実行環境に固有であり，他の環境との識別ができるという性質を示す．個別性が十分に満たされていないサンドボックス情報を検知に用いた場合，検知フェーズでの誤検知が発生してしまうため，検知に用いるサンドボックス情報には高い個別性が求められる．

- （取得行為の）ステルス性

最後に，（取得行為の）ステルス性とは，当該サンドボックス情報を取得している事実を，解析システム側から秘匿できるという性質である．検知の目的で個別性が高いサンドボックス情報にアクセスする場合，ステルス性が十分でない場合は，解析システム側にサンドボックス検知の可能性を推測され，対策をとられる可能性が高くなる．

## 5. 検 証 実 験

本章では実運用されている 15 個の Public MSAS に対して，デコイ挿入攻撃の検証を行う．今回の実験では，ネットワーク関連の情報として IP アドレス，ホスト情報として Windows API である kernel32.dll や advapi32.dll，WMI（Windows Management Instrumentation）を用いて取得が可能な各種のサンドボックス情報（**表 1**）を含めた計 16 種のサンドボックス情報を実験対象とした．

まず，サンドボックス情報暴露フェーズの実験として，16 種類のサンドボックス情報の収集を試みることで，取得安定性と個別性をそれぞれ評価する．収集の際には，解析レポート経由でのサンドボックス情報の収集も行い，その有効性を検証する．

その後，サンドボックス検知フェーズの実験として，暴露フェーズで収集したサンドボックス情報を基に検知が可能であるか検証する．

**表 1**　デコイ（実行ファイル）が取得するサンドボックス情報一覧
Table 1　List of sandbox information collected by a decoy sample.

| サンドボックス情報名 | 取得する情報 |
|---|---|
| arp | 周辺機器のMACアドレス |
| bbn | マザーボードの名前 |
| bbns | マザーボードのシリアル番号 |
| bn | BIOSの名前 |
| bs | BIOSのシリアル番号 |
| cn | コンピュータ名 |
| cpu | CPUの名前 |
| cpuid | CPUのシリアル番号 |
| dn | ディスクドライブの名前 |
| ds | ディスクドライブのシリアル番号 |
| insd | Windowsのインストール日時 |
| mac | MACアドレス |
| pk | Windowsのプロダクトキー |
| pn | Windowsのプロダクト名 |
| un | ユーザ名 |

以下，各フェーズの実験内容について述べる．

### 5.1　サンドボックス情報暴露フェーズ

まず，サンドボックス情報暴露フェーズとして，各 Public MSAS へのデコイ挿入を行い，サンドボックス情報を収集した．さらに，その結果からサンドボックス検知フェーズで用いる情報を選定した．

#### 5.1.1　Public MASA-F へのデコイ（実行ファイル）作成

Public MSAS-F へ投稿するデコイとして以下の挙動を示す検体を作成した．なお，デコイの各インスタンスには識別用の ID が割り振られている．

1. デコイは実行されると，表 1 に示す 15 種のサンドボックス情報を取得する．それらの取得に関しては，Windows API を用いた．なお，IP アドレスに関しては，暴露サーバ側によるアクセスの監視によって取得する．
2. 取得したサンドボックス情報を base64 エンコードし，さらに URL エンコードする．

3. “HKEY_CURRENT_USER” の下にサブキーを作成する．
4. ステップ 3 で作成したサブキーの下にサンドボックス情報名でエントリを作成し，実際に取得したサンドボックス情報を当該エントリの値として設定する．
5. 暴露サーバのドメイン名の名前解決を行い，サーバへの接続を試みる．
6. 暴露サーバとの 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．このとき，各デコイの ID をリクエストファイル名とすることで，暴露サーバ側でデコイの個体識別を行う．さらに，GET リクエストの引数として，ステップ 1 で取得した情報を送信する．

上記のデコイは，レジストリへの書き込みによって解析レポート経由でのサンドボックス情報の暴露を試み，暴露サーバへの通信によってネットワーク経由での暴露を試みる．

#### 5.1.2　Public MSAS-W へのデコイ（URL）の作成

Public MSAS-W へ投稿するデコイとして，暴露サーバのドメイン名と，CGI ファイルの名前を ID として，投稿用 URL を作成した（例：http://check.com/cgi-bin/ID.cgi）．

なお今回の Public MSAS-W に対する実験では，サンドボックス情報として暴露サーバ側で IP アドレスのみを取得することとし，攻撃コードの送信によってデコイ（実行ファイル）をダウンロード・実行させることなどは行わなかった．

#### 5.1.3　暴露サーバの用意

暴露サーバ上に各デコイの ID に対応した CGI ファイルを作成・設置した．暴露サーバは以下の挙動を示す．

1. クライアントからの接続要求を受付ける．
2. 有効な ID を含む HTTP GET リクエストを受信すると，CGI が実行され，事前に用意された無害な Web コンテンツを返信する．
3. CGI はさらに，リクエストの引数を取得し，URL デコード，base64 デコードによりサンドボックス情報を復号したうえでログに記録する．また，HTTP ヘッダ情報やアクセス元 IP アドレスも記録する．

#### 5.1.4　実 験 内 容

実運用中の 8 種の Public MSAS-F，7 種の Public MSAS-W，計 15 個のシステムに対して，1 日 5 つのデコイを 2010 年 11 月 14 日から 7 日間投稿した．

#### 5.1.5　実 験 結 果

実験結果を**表 2** に示す．表には，解析レポートの発行数と暴露サーバへのアクセス数，各サンドボックス情報の暴露結果についてまとめた．なお，システム 3–7 の IP アドレス以外のサンドボックス情報に関しては，ネットワーク経由と解析レポート経由の両方による暴露が成功しており，投稿した際にネットワーク経由での暴露ができなかった場合には，解析レポート経由での暴露結果を用いて，表を作成した．IP アドレス以外のサンドボックス情報の暴露結果については，(取得した値の種類数)/(値を取得できた回数) の形式で表しており，値が複数取得できた情報については，複数取得した値が完全一致した場合のみを同一値と見なした．

- ネットワーク経由でのサンドボックス情報の暴露結果

システム 3–7，9–15 については，デコイの挿入により暴露サーバへのアクセスが観測され，ネットワーク経由でのサンドボックス情報の暴露が成功した．なお，システム 6 ではデコイの投稿回数に比べ暴露サーバへのアクセス数が少ないが，これは暴露サーバのドメイン名の DNS 名前解決が，解析システム内の不具合によって失敗していることが原因だった．逆にシステム 10，11，15 などの Public MSAS-W では，投稿回数よりも暴露サーバへのアクセス数が多くなっているが，その理由は，それらのシステムが検知率を高めることを目的に，投稿された URL に対してバージョンの異なる Web ブラウザで複数回アクセスを行うなどの機能を有しているためである．

一方，システム 1，2，8 では，暴露サーバへのアクセスが観測されないことから，隔離型のサンドボックスを用いていると推定できる．これらのシステムでは，ネットワーク経由でのサンドボックス情報の暴露は行えないが，そもそも隔離型のシステムでは，マルウェアが C&C サーバからの指令などを受け取ることができないため，C&C サーバからの指令に基づく挙動や，新たな実行ファイルのダウンロードといったような挙動を観測できず，十分な解析が行えない．

- 解析レポート経由でのサンドボックス情報の暴露結果

解析レポート経由でのサンドボックス情報の暴露については，システム 1，2 以外の Public MSAS-F では成功した．なお，システム 9–15 の Public MSAS-W については，ネットワーク経由による IP アドレスの取得しか行っていないため，解析レポート経由による暴露結果は載せていない．

一方，システム 1，2 については解析レポート経由でのサンドボックス情報の暴露はできなかった．しかし，システム 1，2 では解析レポートに含まれる検体の挙動情報が他のシステムに比べて少なく，API によるサンドボックス情報の取得自体が失敗している可能性が高い．また，通常の感染ホストではサンドボックス情報が取得できることから，サンドボックス情報を取得できないという事実から攻撃者に解析環境であることを検知される恐れが

**表 2**　サンドボックス情報暴露フェーズの実験結果
Table 2　Summary of experiments in the decoy injection phase.

| システム | 解析対象 | 投稿回数 | 解析レポートの発行数 | 暴露サーバへのアクセス数 | 各サンドボックス情報の暴露結果 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | IPアドレス数 | arp | bbn | bbs | bn | bs | cn | cpu | cpuid | dn | ds | insd | mac | pk | pn | un |
| 1 | File | 35 | 35 | 0 | | | | | | | | | | | | | | | | |
| 2 | File | 35 | 35 | 0 | | | | | | | | | | | | | | | | |
| 3 | File | 35 | 35 | 35 | 8 (7 in a /28 NW) | 1/35 | 0/0 | 0/0 | 1/35 | 0/0 | 10/35 | 1/35 | 0/0 | 1/35 | 0/0 | 1/35 | 35/35 | 1/35 | 1/35 | 0/0 |
| 4 | File | 35 | 35 | 35 | 7 in a /18 NW | 1/35 | 1/35 | 6/35 | 1/35 | 0/0 | 1/35 | 1/35 | 0/0 | 1/35 | 0/0 | 1/35 | 6/35 | 1/35 | 1/35 | 1/35 |
| 5 | File | 35 | 34 | 35 | 1 | 35/35 | 0/0 | 0/0 | 1/35 | 0/0 | 1/35 | 1/35 | 0/0 | 1/35 | 0/0 | 1/35 | 4/35 | 1/35 | 1/35 | 1/35 |
| 6 | File | 35 | 34 | 31 | 25 | 2/35 | 0/0 | 0/0 | 1/35 | 0/0 | 1/35 | 1/35 | 0/0 | 1/35 | 0/0 | 1/35 | 1/35 | 4/35 | 1/35 | 0/0 |
| 7 | File | 35 | 35 | 35 | 10 in a /27 NW | 35/35 | 1/35 | 1/35 | 1/35 | 1/35 | 1/35 | 1/35 | 0/0 | 0/0 | 1/35 | 0/0 | 10/35 | 1/35 | 1/35 | 1/35 |
| 8 | File | 35 | 35 | 0 | | 2/32 | 1/32 | 1/32 | 1/32 | 2/32 | 2/32 | 1/32 | 0/0 | 1/32 | 0/0 | 1/32 | 2/32 | 2/32 | 1/32 | 1/32 |
| 9 | Web | 35 | 35 | 35 | 8 (7 in a /28 NW) | | | | | | | | | | | | | | | |
| 10 | Web | 35 | 35 | 70 | 1 | | | | | | | | | | | | | | | |
| 11 | Web | 35 | 35 | 140 | 3 | | | | | | | | | | | | | | | |
| 12 | Web | 35 | 35 | 35 | 1 | | | | | | | | | | | | | | | |
| 13 | Web | 35 | 35 | 35 | 2 | | | | | | | | | | | | | | | |
| 14 | Web | 35 | 35 | 35 | 4 in a /30 NW | | | | | | | | | | | | | | | |
| 15 | Web | 35 | 35 | 107 | 4 | | | | | | | | | | | | | | | |

ある．

### 5.1.6　検知フェーズで使用するサンドボックス情報の選定

5.1.5 項で示した各サンドボックス情報の暴露結果を，4.3 節で示した 3 つの性質を考慮して，実際に検知フェーズで用いるサンドボックス情報の選定を行う．

- 取得安定性

まず，IP アドレスに関しては，暴露サーバへのアクセスが観測されたシステムについては，システム 6 の DNS 名前解決エラーを除けば，すべて毎回 IP アドレスが取得できている．さらに，システム 5，10–15 に関しては，IP アドレスは 1〜4 種類とアクセス回数に比べて非常に少ない数の IP アドレスのみが取得できた．また，システム 3，7，9 についても，/27 や/28 という狭い範囲の IP アドレスが取得できていることから，比較的取得安定性が高いといえる．

一方 IP アドレス以外のサンドボックス情報に関しては，表 2 を見ると，1 度も値を取得できなかった情報以外は，ほぼ毎回値が取得できている．また，表 2 において，色塗りされたセルは取得された値が 5 種類以下であることを示している．表を見ると，BIOS の名前（bn），コンピュータ名（cn），CPU の名前（cpu），ディスクドライブの名前（dn），インストール日時（insd），Windows のプロダクトキー（pk），Windows のプロダクト名（pn）などは，ほとんどの Public MSAS-F おいて 5 種類以下となっており，高い取得安定性があると考えられる．

- 個別性

個別性に関しては，まず Windows のプロダクトキーや自身の MAC アドレス，IP アドレスといった情報は，各実行環境において基本的に固有の値になると考えられるため，個別性は高いと考えられる．また，周辺機器の MAC アドレス（arp）や OS インストール日時（insd）といった情報も個体識別に利用できるだけの個別性がある可能性がある．実際に今回の実験においては，これらの値が他のシステムと一致することはなかったが，今後，解析システム以外の通常のユーザ環境も含めてこれらの値が十分な個別性を確保しているかどうかを検討する必要がある．

一方，BIOS の名前，CPU の名前，ディスクドライブの名前などの情報は，型番が同じものでは基本的に同一の値になると考えられることから，個別性は低い．実際，今回の実験において，それらの値については異なるシステムで同一の値が得られている場合が散見され，個体識別に用いることは難しいと考えられる．

また，ユーザ名やコンピュータ名などの情報は，今回の実験中では他のシステムと同一の値になることはなかったが，プロダクトキーや MAC アドレスなどとは異なり自由に値を決めることができるため，必ずしも個別性が高いとはいえない．

• （取得行為の）ステルス性

ステルス性の観点から見ると，今回の実験では IP アドレス以外のサンドボックス情報に関しては，すべて Windows API を用いて取得しているため，取得方法によるステルス性の差は特にないと考えられる．ただし，レジストリから Windows のプロダクトキーなどの情報を取得するという挙動は，解析システム検知を行わないマルウェアではあまり見られない挙動であると予想されるため，ステルス性はさほど高くないと考えられる．

一方 IP アドレスに関しては，取得するために必要な挙動はサーバへアクセスを行うのみであり，そのようなネットワーク挙動は通常のプログラムでもよく見られる挙動であるため，ステルス性は高いといえる．

以上の点を考慮して，今回は，IP アドレス，Windows のプロダクトキー，MAC アドレス，インストール日時の 4 種類のサンドボックス情報を基にそれぞれ検知を試みることとした．ただし Public MSAS-W については，今回の実験では，暴露フェーズにおいて IP アドレスのみを取得したため，IP アドレスのみを用いて検知を試みた．

### 5.2　解析システム検知フェーズ

解析システム検知フェーズでは，暴露フェーズで取得したサンドボックス情報を基に，解析システムの検知と解析の回避が可能であるか検証した．

#### 5.2.1　Public MSAS-F へのテスト検体の作成

Public MSAS-F へ投稿するテスト検体を作成した．各テスト検体は検知用の 3 種類のサンドボックス情報（Windows のプロダクトキー，MAC アドレス，インストール日時）のリストを保持している．テスト検体の動作は以下のとおりとなる．

1. テスト検体は実行されると Windows のプロダクトキー，MAC アドレス，インストール日時を取得する．情報の取得には Windows API を用いた．
2. 取得したサンドボックス情報を base64 エンコードし，さらに URL エンコードする．
3. “HKEY_CURRENT_USER” の下にサブキーを作成する．
4. 取得した各サンドボックス情報と自身が保持しているリストとの照合を行う．
5. リストと一致しない場合，一致しなかったサンドボックス情報について，ステップ 3 で作成したサブキーの下にサンドボックス情報名でエントリを作成し，実際に取得したサンドボックス情報を当該エントリの値として設定する．
6. 検知サーバのドメイン名の名前解決を行い，サーバへの接続を試みる．
7. 検知サーバとの 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．このとき，各デコイの ID をリクエストファイル名とすることで，検知サーバ側でデコイの個体識別を行う．さらに，GET リクエストの引数として，ステップ 1 で取得した情報を送信する．
8. サーバからの応答に特定の文字列が含まれていた場合，サーバへの新たなクエリ送信を行う．

テスト検体は，取得した情報と自身が持つリストとの照合を行うことで，ホストベースでの検知を試みる．検知できなかった場合は，レジストリへサンドボックス情報の書き込みを行わせることで，その事実を把握するとともに，その際のサンドボックス情報の値を把握できるようにした．また，取得した情報をクエリに含ませて検知サーバへ送信し，その応答によって挙動を変えることでネットワークベースでの検知を試みる．

#### 5.2.2　Public MSAS-W へのテスト URL の作成

検知サーバのドメイン名と，CGI ファイルの名前を ID として，投稿用のテスト URL を作成した（例：http://www.detect.com/cgi-bin/ID.cgi）．

#### 5.2.3　検知サーバの用意

検知サーバ上に各テスト検体の ID およびテスト URL に対応した CGI ファイルを作成・設置し，検知用のサンドボックス情報（Windows のプロダクトキー，MAC アドレス，インストール日時，IP アドレス）のリストを用意した．検知サーバの挙動は基本的に，5.1.3 項で示した暴露サーバの挙動と同様である．HTTP GET リクエストの引数の値とリストの照合を行い，リストに値がない場合のみ，テスト検体の動作のトリガとなるキーワードを含むファイルもしくは $\mathrm{HTML}_{Malicious}$ を送信し，リストと一致した場合は無害な Web コンテンツを返す．今回の実験では $\mathrm{HTML}_{Malicious}$ にも無害な Web コンテンツを用いた．

#### 5.2.4　実 験 内 容

実運用中の 6 種の Public MSAS-F，7 種の Public MSAS-W，計 13 種類のシステムに対して，1 日 5 つのテスト検体を 2010 年 11 月 23 日から 2 日間投稿した．なお今回の実験では，暴露フェーズでサンドボックス情報の取得ができなかったシステム 1，2 については，

**表 3**　サンドボックス検知フェーズの実験結果
Table 3　Summary of experiments in the sandbox detection phase.

| システム | 解析対象 | インターネット接続性 | 投稿回数 | 解析レポートの発行数 | 検知サーバへのアクセス数 | ホストベースの検知 | | | ネットワークベースの検知 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | pk | mac | insd | pk | mac | insd | IPアドレス |
| 1 | File | 隔離型 | | | | | | | | | | |
| 2 | File | 隔離型 | | | | | | | | | | |
| 3 | File | インターネット接続型 | 10 | 10 | 10 | 10/10 | 0/10 | 10/10 | 10/10 | 0/10 | 10/10 | 7/10 |
| 4 | File | インターネット接続型 | 10 | 10 | 10 | 10/10 | 10/10 | 10/10 | 10/10 | 10/10 | 10/10 | 0/10 |
| 5 | File | インターネット接続型 | 10 | 10 | 10 | 10/10 | 10/10 | 10/10 | 10/10 | 10/10 | 10/10 | 10/10 |
| 6 | File | インターネット接続型 | 10 | 10 | 10 | 6/10 | 10/10 | 10/10 | 6/10 | 10/10 | 10/10 | 5/10 |
| 7 | File | インターネット接続型 | 10 | 10 | 10 | 10/10 | 10/10 | | 10/10 | 10/10 | | 10/10 |
| 8 | File | 隔離型 | 10 | 10 | 0 | 10/10 | 10/10 | 10/10 | | | | |
| 9 | Web | インターネット接続型 | 10 | 10 | 10 | | | | | | | 4/10 |
| 10 | Web | インターネット接続型 | 10 | 10 | 20 | | | | | | | 20/20 |
| 11 | Web | インターネット接続型 | 10 | 10 | 40 | | | | | | | 40/40 |
| 12 | Web | インターネット接続型 | 10 | 10 | 10 | | | | | | | 10/10 |
| 13 | Web | インターネット接続型 | 10 | 10 | 10 | | | | | | | 10/10 |
| 14 | Web | インターネット接続型 | 10 | 10 | 10 | | | | | | | 10/10 |
| 15 | Web | インターネット接続型 | 10 | 10 | 20 | | | | | | | 20/20 |

検知フェーズの実験は行わなかった．

#### 5.2.5 実験結果

**表 3** に検知フェーズの実験結果を示す．色塗りしたセルは検知が毎回成功したものを示している．

- ホストベースでの検知結果

表 3 を見ると，実験を行った Public MSAS-F では，Windows のプロダクトキー，MAC アドレス，インストール日時を用いた検知結果のほぼすべてで高い検知率になっている．ただし，システム 6 における Windows プロダクトキーによる検知，システム 7 におけるインストール日時による検知では検知できない場合があり，システム 3 における MAC アドレスによる検知は 1 度も成功しなかった．

まず，システム 3 については暴露フェーズの実験も含めて，取得するたびに異なる MAC アドレスが取得できていた．これは，その他のサンドボックス情報と比べて明らかに特徴が異なっている．このことから，システム 3 では解析検知の対策として，解析のたびに MAC アドレスを変更している，もしくは API フックによって API の返り値を変更している可能性がある．

次に，システム 6 では，10 回の投稿中 4 回の投稿時に，暴露フェーズで取得したものとは異なる 1 種類の Windows プロダクトキーが取得され検知ができなかった．これは，暴露フェーズにおいてシステムの用いるすべての Windows プロダクトキーを抜き出せていなかったと考えられるが，暴露フェーズの期間をより長くとれば取得できたと考えられる．

最後に，システム 7 ではつねにインストール日時は取得できなかった．

- ネットワークベースでの検知結果

次にネットワークベースでの検知結果を見ると，まず，システム 3–7 では，IP アドレス以外による検知結果はホストベースでの検知結果と同様であった．システム 8 は隔離型のサンドボックスのため，ネットワークベースでの検知は行っていない．

一方，IP アドレスによる検知に関しては，システム 5，7，10–15 については，すべて検知できた．ただし，システム 3，6，9 では検知できない場合があり，システム 4 では，1 度

も検知ができなかった．

システム 3，9 については，新たに観測された IP アドレスは，暴露フェーズで観測された IP アドレスと同じ/28 のサブネットの IP アドレスであったため，その範囲の IP アドレスを検知リストとして用いることで容易に検知が可能である．

次に，システム 4 については，約 1 日置きに/18 のサブネットの範囲で IP アドレスが変わっており，その結果 1 度も検知ができなかった．その IP アドレスを調査したところ，とある ISP に割り当てられている IP アドレス範囲の一部であることが分かった．このことからシステム 2 では，インターネットとの接続に ISP 回線を利用し，おそらくサンドボックスの IP アドレスを特定されないために，約 1 日おきに PPPoE セッションを張り替えるなどして割り当てられる IP アドレスを変更していると推定される．

最後に，システム 6 については，国も地域も異なる多数の IP アドレスからアクセスが観測された．これらの IP アドレスについて調査したところ，IP アドレスの多くが Onion ルータの IP アドレスリスト[40] に含まれていた．このことから，システム 6 ではサンドボックスの IP アドレスを特定されないために，サンドボックスをインターネットに接続する際に匿名通信路である Tor[7] を利用していると考えられる．

### 5.3　実験結果まとめ

実験により，既存の多くの Public MSAS-F で，サンドボックス情報を解析レポート経由で暴露させる手法が有効であることが示せた．この手法を用いることで，ネットワーク経由での暴露ができない隔離型のシステムに対しても，サンドボックス情報を取得することが可能であることが明らかになった．

また，取得したサンドボックス情報を用いた検知に関しても，先行研究で実証した IP アドレスによる検知だけではなく，Windows プロダクトキーや MAC アドレス，インストール日時などのサンドボックス情報を用いた場合でも Public MSAS による解析を回避することが可能であることが明らかになった．加えて，いくつかのシステムでは，IP アドレスや MAC アドレスといった特定のサンドボックス情報に関して検知対策を行っていることが推定された．ただし，それらのシステムにおいても別のサンドボックス情報を用いることで検知が可能であり，デコイ挿入攻撃に対する対策は十分ではないことが明らかになった．

## 6. 考　　察

### 6.1　他の解析システム検知手法との比較

2 章の関連研究で述べた仮想化システム検知やデバッガ検知などを用いた解析システム検知手法とデコイ挿入攻撃を比較した場合，前者は攻撃者が個々の解析システムについて対応しなくても，それらを利用する解析システム全般を検知できる点が利点としてあげられる．しかし，一方で仮想化システムなどを活用するのは解析システムに限らないため，通常のユーザ環境を解析システムだと誤検知してしまう可能性がある点が欠点である．

次に，後者のデコイ挿入攻撃について考えると，デコイ挿入攻撃ではある特定の解析システムのサンドボックス情報を基に検知を行うため，（個別性が十分に高いサンドボックス情報を用いるならば）通常のユーザ環境を誤検知してしまう可能性は低くなる．また，近年では Public MSAS の人気が高まり多くのユーザが Public MSAS を利用している背景から，特定の解析システム（Public MSAS）による解析を回避できることが攻撃者にとって大きな利点になってきていると考えられる．しかし，特定の解析システムを検知対象としているため，攻撃者がその存在を把握していない解析システムに関しては検知ができないという点が欠点としてあげられる．

### 6.2　解析レポートを介したサンドボックス情報の流出

先行研究では，ネットワーク経由での暴露方法のみしか実証していなかったが，今回の実験結果から，隔離型のサンドボックスかインターネット接続型のサンドボックスにかかわらず，解析レポート経由でサンドボックス情報を暴露させることが可能であることが実証できた．このような解析レポート経由での情報流出に対する最も基本的な対策は，解析レポートに含めるマルウェアの挙動情報を減らすことであるが，これはユーザにとって Observability を下げることとなるため注意が必要である．

### 6.3　IP アドレス以外のサンドボックス情報を用いた解析環境検知

今回の実験から，IP アドレス以外でも Windows プロダクトキーや MAC アドレス，インストール日時といったサンドボックス情報を用いることで，解析システムの検知が可能であることを示せた．このことから，Public MSAS を設計する際は，IP アドレスの流出だけでなく，サンドボックスを特定されるようなシステム情報の流出についても適切な対策が必要といえる．この際，サンドボックス情報の取得を完全に妨害するような対策は適切ではない．なぜなら，攻撃者はサンドボックス情報が取得できない事実をもって解析環境検知を行うことが想定されるからである．そのため，解析を行うごとにサンドボックス情報を変更したり，サンドボックス情報を取得するための API をフックし，値を偽装したりするなどの対策が必要といえる．実際に，システム 3 では MAC アドレスの値については実行ごとに異なる値を設定するという対策をとっていると推定できるが，その他の情報については対策が行われておらず，攻撃への対策が十分ではない．

### 6.4 IP アドレス暴露への対策

IP アドレスの暴露への対策としては，まず単純に ISP 回線などを用いて頻繁に PPPoE セッションを張り替え，サンドボックスの IP アドレスを変更する対策が考えられる．実際，システム 4 では実験結果から，そのような対策をとっていると推定できる．その結果，システム 4 では今回の実験の範囲では IP アドレスによる検知は行えなかったが，頻繁にデコイを投稿しサンドボックス情報を取得することで攻撃者に対応されてしまう可能性がある．

また，匿名通信路を用いて IP アドレスを隠蔽する手法も考えられる．実際，システム 6 では，実験結果で述べたように匿名通信路である Tor の Onion ルータの IP アドレスからのアクセスが観測され，上記の対策手法をとっていると推定される．このような Tor の利用は解析システムの IP アドレスの隠蔽を可能とするが，その場合でも Tor によるアクセスかどうかを判定することは可能である[5),44)]．通常マルウェアに感染するような一般ユーザの大多数は Tor を利用しているとは考えにくいため，Tor を利用しているという事実から攻撃者に解析環境を検知されてしまう可能性がある．

また，我々は論文 19) で投稿者のマシンをプロキシとして用いることで，サンドボックスの IP アドレスを隠蔽する手法を提案しているが，サンドボックス内で動作するマルウェアからの通信が投稿者のマシンを経由して実インターネットに送られるという仕組みは投稿者にとって心理的抵抗感があると思われ，提供可能性の検討は今後の課題である．

### 6.5 Public MSAS 以外へのデコイ挿入攻撃

本稿では，Public MSAS を対象としたデコイ挿入攻撃について述べたが，このようなデコイを挿入する対象は Public MSAS に限らない．たとえばオンラインスキャンサービス[48)]へ投稿したり，検体共有サイト[37)]に投稿したりすることもできる．その結果，それらのコミュニティにおける検体共有の流れや，非公開の解析システムの情報も観測できる可能性がある．そのため，解析者側は，このようなデコイ挿入攻撃の危険性を適切に把握し，対策を行うとともに，検体や各種情報の共有についても注意深く行う必要がある．

## 7. ま と め

本稿では，公開型マルウェア動的解析システムへの攻撃手法として我々が提案したデコイ挿入攻撃について，先行研究では実証しきれていなかった，解析レポートを介したサンドボックス情報の暴露や，IP アドレス以外のサンドボックス情報を用いた攻撃が，既存の多くの解析システムに対して有効であることを示した．その結果，先行研究で必要性を指摘した IP アドレス暴露への対策のみでは，デコイ挿入攻撃への対策としては不十分であり，IP アドレス以外のサンドボックス情報による検知への対策や，解析レポート経由による暴露への対策などを含めた，総合的なデコイ挿入攻撃への対策が必要であることが分かった．

## 参 考 文 献


1) Bailey, M., Oberheide, J., Andersen, J., Mao, Z.M., Jahanian, F. and Nazario, J.: Automated Classification and Analysis of Internet Malware, *Proc. Recent Advances in Intrusion Detection*, *RAID'07*, LNCS Vol.4637, pp.178–197 (2007).
2) Bayer, U., Comparetti, P.M., Hlauschek, C., Kruegel, C. and Kirda, E.: Scalable, Behavior-Based Malware Clustering, *Symposium on Network and Distributed System Security* (*NDSS*) (2009).
3) Bayer, U., Habibi, I., Balzarotti, D., Kirda, E. and Kruegel, C.: A View on Current Malware Behaviors, *Proc. 2nd Usenix Workshop on Large-Scale Exploits and Emergent Threats*, *LEET'09* (2009).
4) Bayer, U., Kruegel, C. and Kirda, E.: TTAnalyze: A Tool for Analyzing Malware, *15th Annual Conference of the European Institute for Computer Antivirus Research* (*EICAR*) (2006).
5) Chakravarty, S., Stavrou, A. and Keromytis, A.D.: Identifying Proxy Nodes in a Tor Anonymization Circuit, *Proc. 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp.633–639 (2008).
6) Chen, X., Andersen, J., Mao, Z.M., Bailey, M. and Nazario, J.: Towards an Understanding of Anti-virtualization and Antidebugging Behavior in Modern Malware, *The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (*DSN 2008*), pp.177–186 (2008).
7) Dingledine, R., Mathewson, N. and Syverson, P.: Tor: The Second-Generation Onion Router, *Proc. 13th USENIX Security Symposium*, pp.303–320 (2004).
8) Holz, T. and Raynal, F.: Detecting Honeypots and other Suspicious Environments, *Proc. 2005 IEEE Workshop on Information Assurance and Secrity*, pp.29–36 (2005).
9) Inoue, D., Yoshioka, K., Eto, M., Hoshizawa, Y. and Nakao, K.: Malware Behavior Analysis in Isolated Miniature Network for Revealing Malware's Network Activity, *IEEE International Conference on Communications* (*ICC 2008*), pp.1715–1721 (2008).
10) Inoue, D., Yoshioka, K., Eto, M., Hoshizawa, Y. and Nalao, K.: Automated Malware Analysis System and its Sandbox for Revealing Malware's Internal and External Activities, *IEICE Trans.*, Vol.E92D, No.5, pp.945–954 (2009).
11) Kirda, E., Kruegel, C., Banks, G., Vigna, G. and Kemmerer, R.: Behavior-based Spyware Detection, *Proc. 15th Conference on USENIX Security Symposium*, Vol.15,

No.19, pp.273–288 (2006).
12) Miwa, S., Miyachi, T., Eto, M., Yoshizumi, M. and Shinoda, Y.: 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*, p.6 (2007).
13) Porras, P., Saidi, H. and Yegneswaran, V.: A Foray into Conficker's Logic and Rendezvous Points, *Proc. USENIX Workshop on Large-Scale Exploits and Emergent Threats*, p.7 (2009).
14) Raffetseder, T., Kruegel, C. and Kirda, E.: Detecting System Emulators, *Proc. 10th Information Security Conference* (*SC*), LNCS Vol.4779, pp.1–18 (2007).
15) Rajab, M.A., Zarfoss, J., Monrose, F. and Terzis, A.: A Multifaceted Approach to Understanding the Botnet Phenomenon, *Proc. 6th ACM SIGCOMM Conference on Internet Measurement*, pp.41–52 (2006).
16) Wang, P., Wu, L., Cunningham, R. and Zou, C.C.: Honeypot Detection in Advanced Botnet Attacks, *International Journal of Information and Computer Security 2010*, Vol.4, No.1, pp.30–51 (2010).
17) Willems, C., Holz, T. and Freiling, F.: Toward Automated Dynamic Malware Analysis Using CWSandbox, *Security & Privacy Magazine*, Vol.5, Issue 2, pp.32–39, IEEE (2007).
18) Yoshioka, K., Inoue, D., Eto, M., Hoshizawa, Y., Nogawa, H. and Nakao, K.: Malware Sandbox Analysis for Secure Observation of Vulnerability Exploitation, *IEICE Trans.*, Vol.E92D, No.5, pp.955–966 (2009).
19) 吉岡克成，細渕嘉彦，織井達憲，松本　勉：マルウェア動的解析オンラインサービスの脆弱性，情報処理学会コンピュータセキュリティシンポジウム 2009（CSS2009），F5-1 (2009).
20) Yoshioka, K. and Matsumoto, T.: Multi-pass Malware Sandbox Analysis with Controlled Internet Connection, *IEICE Trans.*, Vol.E93A, No.1, pp.210–218 (2010).
21) Yoshioka, K., Hosobuchi, Y., Orii, T. and Matsumoto, T.: Your Sandbox is Blinded: Impact of Decoy Injection to Public Malware Analysis Systems, *IPSJ Journal*, Vol.52, No.3 (2011) (accepted).
22) Agues, available from ⟨http://www.aguse.jp⟩.
23) Anubis, available from ⟨http://analysis.seclab.tuwien.ac.at/⟩.
24) Autovin, available from ⟨http://autovin.pandasecurity.my/⟩.
25) BitBlaze, available from ⟨https://aerie.cs.berkeley.edu/⟩.
26) Comodo Instant Malware Analysis, available from ⟨http://camas.comodo.com/cgi-bin/submit⟩.
27) CWSandbox, available from ⟨http://www.cwsandbox.org/⟩.
28) Dr. Web online check, available from ⟨http://online.us.drweb.com/?url=1⟩.
29) Eckersley, P.: How Unique is Your Web Browser?, available from ⟨https://panopticlick.eff.org/browser-uniqueness.pdf⟩.
30) Fingers, E.: Sandbox Awareness, available from ⟨http://evilfingers.blogspot.com/2009/01/sandbox-awareness.html⟩.
31) Evilcodecave's Weblog, OffensiveCOding updated – Emulation/AV Awareness, available from ⟨http://evilcodecave.wordpress.com/tag/cwsandbox/⟩.
32) gred, available from ⟨https://www.gred.jp/?tab=goleo⟩.
33) Joebox, available from ⟨http://www.joebox.org/⟩.
34) Jsunpack, available from ⟨http://jsunpack.jeek.org/dec/go⟩.
35) LinkScanner, available from ⟨http://linkscanner.explabs.com/linkscanner/default.aspx⟩.
36) Norman Sandbox, available from ⟨http://www.norman.com/security_center/security_tools/⟩.
37) Offensive Computing, Community Malicious code research and analysis, available from ⟨http://www.offensivecomputing.net⟩.
38) Online Link Scan, available from ⟨http://onlinelinkscan.com⟩.
39) OpenSC.ws: Detect 5 different sandboxes, available from ⟨http://www.opensc.ws/snippets/3558-detect-5-different-sandboxes.html⟩.
40) Proxy.org – Tor servers – Tor IP List, available from ⟨http://proxy.org/tor.shtml⟩.
41) Rutkowska, J.: Red Pill .. or how to detect VMM using (almost) one CPU instruction, available from ⟨http://invisiblethings.org/papers/redpill.html⟩.
42) Sandboxie, available from ⟨http://www.sandboxie.com⟩.
43) Sunbelt CWSandbox, Malware Research Labs, available from ⟨http://www.sunbeltsecurity.com/⟩.
44) Tor or not Tor: How to tell if someone is coming from a Tor exit node, in PHP, available from ⟨http://www.irongeek.com/i.php?page=security/detect-tor-exit-node-in-php⟩.
45) Threat Experts, available from ⟨http://www.threatexpert.com/⟩.
46) Unmask Parasite, available from ⟨http://www.unmaskparasites.com/⟩.
47) viCHECK.ca, available from ⟨https://vicheck.ca/⟩.
48) Virustotal, available from ⟨http://www.virustotal.com/⟩.
49) Webutation, available from ⟨http://www.webutation.org/⟩.
50) Wepawet, available from ⟨http://wepawet.iseclab.org/⟩.
51) vURL Online, available from ⟨http://vurldissect.co.uk/⟩.
52) Win32.agobot, available from ⟨http:/www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=37776⟩.

(平成 22 年 12 月 2 日受付)
(平成 23 年 6 月 3 日採録)

## 推　薦　文

ユーザからファイル検体を受付けて解析環境内で挙動を分析し，結果をユーザに提供するサービスが，マルウェア対策の 1 つとして注目されている．本論文は，当該サービスの脆弱性を指摘し体系的にまとめるとともに，実験による検証と考察を丁寧に行っている．技術的貢献度および資料価値のどちらも高いので，推薦したい．

（コンピュータセキュリティシンポジウム 2010 プログラム委員長　松浦幹太）


**笠間　貴弘**（学生会員）

2011 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程前期修了，修士（工学）．同年 4 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期に進学．同年 4 月より独立行政法人情報通信研究機構で研究員として勤務．マルウェア解析やネットワーク攻撃観測等のネットワークセキュリティの研究に従事．

**織井　達憲**

2011 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻情報メディア学コース博士課程前期修了，工学修士．マルウェア解析等のネットワークセキュリティに関する研究に従事．2011 年 4 月より日本電気株式会社に勤務．

**吉岡　克成**（正会員）

2005 年 3 月横浜国立大学大学院環境情報学府情報メディア環境学専攻博士課程後期修了，博士（工学）．同年 4 月独立行政法人情報通信研究機構研究員．2007 年 12 月より横浜国立大学学際プロジェクト研究センター特任教員（助教）．2011 年 4 月より横浜国立大学大学院環境情報研究院准教授．マルウェア解析やネットワーク攻撃観測・検知等のネットワークセキュリティの研究に従事．2009 年文部科学大臣表彰・科学技術賞（研究部門）受賞．

**松本　　勉**（正会員）

1986 年 3 月東京大学大学院工学系研究科電子工学専攻博士課程修了，工学博士．同年 4 月横浜国立大学講師．2001 年 4 月より同大学院環境情報研究院教授．2007 年 4 月〜2011 年 3 月は同大学教育研究評議員を兼務．2011 年 4 月より同大学理工学部副学部長を兼務．日本学術会議連携会員．暗号アルゴリズム・プロトコル，耐タンパー技術，生体認証，人工物メトリクス等の「情報・物理セキュリティ」の研究教育に 1981 年より従事．1982 年にオープンな学術的暗号研究を目指した「明るい暗号研究会」を 4 名で創設．2005 年〜2010 年国際暗号学会 IACR 理事．1994 年第 32 回電子情報通信学会業績賞，2006 年第 5 回ドコモ・モバイル・サイエンス賞，2008 年第 4 回情報セキュリティ文化賞，2010 年文部科学大臣表彰・科学技術賞（研究部門）受賞．