情報処理学会論文誌　Vol. 50　No. 9　2127–2136 (Sep. 2009)

セキュリティ無効化攻撃を利用したマルウェアの検知と活動抑止手法の提案

松木隆宏[†1]　新井悠[†1]
寺田真敏[†2]　土居範久[†2]

近年のマルウェアは，感染活動や悪意のある活動を隠蔽する機能を備える傾向が強まっている．活動を隠蔽する機能には，対策に必要な解析を妨げる耐解析機能や，OSのセキュリティ機能，ウイルス対策ソフトウェアやパーソナルファイウォールを無効化する機能など多岐にわたり，マルウェア対策もこれらの機能をふまえた対応が必要となってきている．本論文では，セキュリティ無効化攻撃の代表的な手法であるプロセスの強制終了について，マルウェアの検体およびインターネットから入手したマルウェアのソースコードの調査結果を報告する．その調査結果に基づき，プロセス強制終了攻撃を誘発させるおとりのプロセスを用いて，マルウェアを検知し，マルウェアの活動を停止させるプロアクティブな手法を提案する．また，実装したプロトタイプシステムとハニーポットで収集した検体を用いて提案手法の有効性を示す．

Proposal of Anti-Malware Technique Turning Security Disabling Attack to Advantage

Takahiro Matsuki,[†1] Yuu Arai,[†1] Masato Terada[†2] and Norihisa Doi[†2]

Latest malware tend to be designed infect secretly and conceal their activity. Stealth feature includes anti-reverse engineering function and hinder OS's security function, anti-virus program, personal fire wall, and so on. Countermeasure should be considering them. In this paper, we report the result of the survey about malware's characteristic security disablement feature, forcible process killing and malware's source code detail. Based on survey results, we offered decoy processes which proactively counter forcible security program shut down. Moreover verify effectivity of proto type system using the specimen which gathered with honeypots.

1. はじめに

インターネット常時接続の普及と利用形態の多様化にともなって，ネットワークに蔓延するマルウェアも多種多様になり，その被害が増加している．特に，攻撃者からの命令に従ってスパムメール配信やDDoS攻撃などの攻撃活動を行い，さらに，自身のアップデートや別のマルウェアをダウンロードする機能を備えるボット形態のマルウェアの増加は顕著である．

これらボットの中には，動的解析を回避するデバッガ検知，静的解析による特徴抽出やデータ抽出を妨げる暗号化や難読化，自身の活動を阻害するセキュリティソフトウェアなどを強制終了するセキュリティ機能の無効化など，自己を守るための機能，自己防衛機能を備えているものが存在している[1)]．

本論文では，マルウェアの自己防衛機能の1つである，セキュリティ機能を無効化する攻撃（以降，セキュリティ無効化攻撃）に着目し，マルウェアの活動を抑止する新しい対策手法を提案する．まず，マルウェアの攻撃対象となるシステム上におとりのプロセスを用意する．次に，おとりプロセスを強制終了させようとするプロセスを検知した場合，そのプロセスをマルウェアと判断し停止させることで，マルウェアの活動を抑止する手法である．

本論文の構成は次のとおりである．まず，2章においてマルウェアの自己防衛機能の分類，各種の自己防衛機能の例と関連研究の概況について述べる．3章で，インターネットで流通しているボットのソースコードの調査結果に基づき，自己防衛機能の1つであるセキュリティ無効化攻撃の詳細を示す．4章では，調査結果で得られたセキュリティ無効化攻撃に着目し，これを利用した新たなマルウェア検知および活動抑制手法の提案と実装方法について述べる．そして，5章では，提案手法を実現するプロトタイプシステムとハニーポットで収集したマルウェア検体を用いて評価し，その有効性を示す．

2. 関連研究

近年，自己を守るための機能，自己防衛機能を備えているマルウェアが増加している．自己防衛機能は，以前から知られている機能の複合化や個々の機能の高度化によって巧妙化が

†1 株式会社ラック サイバーリスク総合研究所
Risk Research Institute of Cyber Space, Little eArth Corporation Co., Ltd.
†2 中央大学
Chuo University

進んでいる．本章では，マルウェアの自己防衛機能に関する研究の概況について述べる．

(1)　動的解析の妨害

マルウェアの実行を通した動的解析を妨げるための手法である．デバッガによる解析，あるいは，仮想マシン環境での実行を検知すると停止するように実装されている．

- デバッガ検知（Anti-Debugging）
  デバッガを利用した動的解析を妨げることを目的とし，デバッガを検知した場合にはマルウェア自身を停止するなどの動作をとる．RBOT，SDBOT，IRCBOT，SPYBOT などの著名なボットで確認されている．関連研究として，マルウェアのデバッガ検知機能を無効化してマルウェア解析を効率化する手法の研究が行われている[2),3)]．また，著者らは，デバッガ検知を逆に利用して，デバッガ検知機能を意図的に動作させてマルウェアの活動を抑止する手法を報告している[4)]．
- 仮想マシン検知
  仮想マシン環境を利用した動的解析を妨げることを目的とし，仮想マシン環境で実行されていることを検知した場合に，マルウェア自身を停止するなどの動作をとる．これまで，VMware の独自インタフェースを利用したバージョン情報取得による検知手法が，W32/Agobot.hm で確認されている．また，Packer が提供する仮想マシン検知機能を利用したマルウェアも存在する[⋆1]．

(2)　静的解析の妨害（難読化・パッキング）

ペイロードに暗号化などを施し，逆アセンブルによる特徴抽出，データ抽出などの静的解析を妨げる手法である．暗号化だけではなく，UPX[5)] などの実行可能ファイルの圧縮を組合せた難読化も行われている．たとえば，W32/MSBlaster も UPX による難読化，圧縮が施されていた．近年，特に Packer が提供する複数の耐解析機能と高度な難読化を利用したマルウェアが増加している．こうしたパッキングされたマルウェアの効率的な解析手法の研究として，メモリアクセスの監視による自動アンパック手法が研究されている[6)]．

(3)　ステルス化

マルウェア自身が存在していることを隠したり，実行されている事実を隠したり，あるいは，ファイルやレジストリ，ブートセクタなどへの改変を隠したりするなど，侵害活動に関わる情報を隠す手法である．この機能を備えたマルウェアは，2000 年に W95/Smash が発見されている．2007 年に入ってからは Storm Worm のように，ルートキット技術を利用してステルス化を行うマルウェアが増加している[7)]．

(4)　セキュリティ機能の無効化

本論文で着目したマルウェアの自己防衛機能である．マルウェアの活動を阻害する各種セキュリティソフトウェアの動作を無効化し，能動的に自己を防衛する手法である．マルウェアの検知および削除を行うウイルス対策ソフトウェアや外部との通信のアクセス制御を行うファイアウォールを攻撃の対象とする．

- データファイルの削除
  ウイルス対策ソフトウェアに必要な特定のデータファイルを探して削除することにより，マルウェアの検知を妨害する．1998 年に W32/Marburg というマルウェアが特定のファイルを削除することが確認されている．
- 対策情報の遮断
  ウイルス対策ソフトウェアのパターンファイルなどの対策情報の更新を妨害する手法である．具体的には，ウイルス対策ベンダのサイトのドメイン名をリスト化し，該当するサイトへのアクセスを遮断する．2004 年 3 月に発見された W32/Agobot.hm は，Windows[⋆2] システムのネットワーク設定ファイルである hosts ファイルに，symantec.com，sophos.com などに対応する IP アドレスとして，ループバックアドレスやブロードキャストアドレスを設定し，ウイルス対策ベンダのサイトへのアクセスを遮断する．
- プロセスの強制終了
  ウイルス対策ソフトウェアやファイアウォールなどのセキュリティソフトウェアのプロセスを見つけた場合，そのプロセスを強制的に終了する．2002 年 9 月に発見された W32/Bugbear は，ウイルス対策ソフトウェアなどセキュリティ関連のプロセス名（Zonealarm.exe，Webscanx.exe，Navw32.exe など）を検索し，強制終了する．著者らは，おとりのプロセスを利用して，プロセスの強制終了を試みるマルウェアを検知し，逆にマルウェアの活動を抑止する手法について報告している[8)]．

これまで，各種の自己防衛機能に関する調査や研究が，いくつか行われているが，自己防

⋆1 商品名称などに関する表示
VMware は，米国 VMware, Inc. の米国およびその他の国における登録商標または商標です．

⋆2 Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です．

衛機能を逆に利用して対策につなげるというアプローチの研究はまだ少ない．本研究では，マルウェアが自己防衛機能を備えてきていることをふまえ，自己防衛機能を逆に利用する手法を確立し，マルウェアによる侵害活動を抑制することを目的とする．

本論文では，セキュリティソフトウェアを強制終了するマルウェアによるセキュリティ無効化攻撃に着目し，プロセス強制終了を誘発させるおとりのプロセスを利用して，マルウェアの検知および活動を抑止する手法を提案する．

3. セキュリティ無効化攻撃の調査

一般的な Windows システムにおけるセキュリティ対策として，ウイルス対策ソフトウェアやパーソナルファイアウォールなどのセキュリティソフトウェアが導入されている場合が多い．近年のマルウェアの多くは，感染対象のシステムにウイルス対策ソフトウェアなどが存在することを前提に作成されており，感染対象のセキュリティ機能を無効化あるいは低下させて，自身の存在や活動を隠蔽しつつ，悪意のある活動を行うと考えられる．

本章では，マルウェアのセキュリティ無効化攻撃の 1 つであるプロセスの強制終了に関する調査結果について述べる．

3.1 調 査 方 法

調査は，インターネットから入手したマルウェアのソースコードの解析とマルウェア検体の静的解析によって行った．ソースコードの解析は，マルウェアの動作を正確に解明する有効な方法である．また，ソースコードが入手可能なマルウェアは，それを基にした亜種が多数作成されているため，亜種間に共通する特徴の抽出にも有効である．調査したマルウェアのソースコードのアーカイブ名および検体名称を次に示す．

(1)　マルウェアのソースコード（7 種）

- Rxbot-IFS1.1.priv
- RxWOLF-svchost2Universal
- rBot-lsass
- Rxbot-EcLiPsE1.1.priv
- Rose v1.3 2007 byDreamWoRK
- N1c0
- Sdnbbot_sp2mod_wks_kelvin

(2)　マルウェア検体（1 種）

- WORM_ALLAPLE.AC

3.2 調 査 結 果

本節では，調査の結果判明したマルウェアによる OS のセキュリティ機能の無効化とセキュリティソフトウェアの無効化の概要について述べる．

3.2.1 OS のセキュリティ機能の無効化

Windows XP SP2 以降の Windows には，セキュリティを強化する機能として，システムの自動更新，Windows ファイアウォールおよび，ウイルス対策ソフトウェアの動作を監視する機構が搭載されている．このように OS が提供するセキュリティ機能を OS のセキュリティ機能と呼ぶ．

マルウェア検体（WORM_ALLAPLE.AC）の静的解析の結果，システムコマンドやユーティリティを介して，これら Windows に搭載されている OS のセキュリティ機能を無効化する処理が存在することを確認した．具体的には，マルウェア検体は Windows ファイアウォールなどの起動/停止システムコマンドを実行するバッチファイルを生成した後，そのバッチファイルの実行を通して OS のセキュリティ機能を無効化する．

このように，システムコマンドを利用して無効化する理由としては，ウイルス対策ソフトウェアによる振舞い検知の回避などがあげられる．

3.2.2 セキュリティソフトウェアの無効化

2007 年初旬にインターネットから入手したボットなどのマルウェアのソースコード 7 種類を調査した結果，これらソースコードにおいて，セキュリティソフトウェアの無効化する処理が実装されていることを確認した．いずれも，セキュリティソフトウェアの無効化にあたり，3.2.3 項で述べるウイルス対策ソフトウェアやシステムモニタリングツールなどのセキュリティソフトウェアのプロセス名のリストを持っており，システム上で実行されているプロセスが，リストにあるプロセス名リストと一致した場合，そのプロセスを強制終了する．

このような強制終了を実現するにあたり，マルウェアは，自身にシステムプロセスのデバッグを可能にする特権を付与した後，対象プロセスを強制的に終了するという方法をとる．これにより，対象となるプロセスがマルウェアによる強制終了に対する防御機能を備えていない場合，プロセスを容易に停止させることができてしまう．マルウェアが対象のプロセスを停止させるまでの Windows API の呼び出しの流れは，次のとおりである．

(1)　デバッグ特権の取得

マルウェアは，OpenProcessToken API により自身のプロセスのアクセストークンを開き，AdjustTokenPrivileges API によりシステムプロセスをデバッグするための特権 SeDebugPrivilege を有効にする．

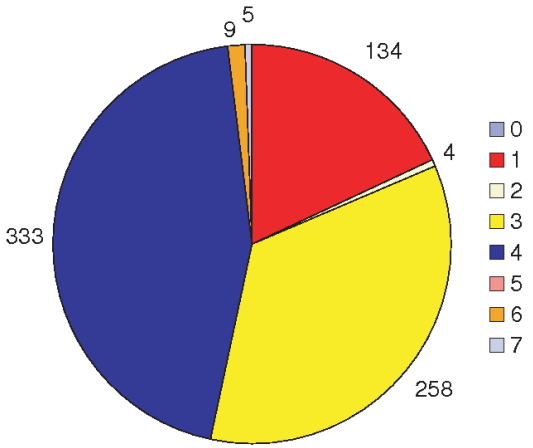

図 1　プロセス名共通度の割合
Fig. 1　Common degree of targeted process names.

(2)　停止対象プロセス名の探索
CreateToolhelp32Snapshot API によりシステム上で実行されているプロセスのスナップショット（一覧）を作成し，Process32First API，Process32Next API を用いて強制終了対象プロセス名とシステム上で実行されているプロセス名との比較を行いながら，停止対象プロセス名を探索する．なお，調査した範囲では，停止対象プロセスの判定はプロセス名のみで，実行ファイルのパスやミューテックスを用いた確認は行っていない．

(3)　対象プロセスの強制終了
上記の探索において特定したプロセスのハンドルを OpenProcess API で開き，TerminateProcess API で指定されたプロセスおよびそのプロセスに属するすべてのスレッドを強制的に終了させる．

3.2.3　強制終了対象となるプロセス名

本項では，マルウェアが強制終了対象とするプロセス名の傾向について示す．

7 種類のソースコードには，強制終了の対象となるプロセス名のリストがハードコーディングされており，合計で 743 種類のプロセス名が存在した．それぞれのマルウェアが保持するプロセス名の共通度を**図 1** に示す．

表 1　強制終了されるソフトウェアのカテゴリ
Table 1　Targeted software category.

種別	件数
セキュリティソフトウェア	191
システムコマンド，ユーティリティ	49
ウイルス，ワーム，トロイの木馬	31
詳細不明	76
合計	347

共通度 4 以上（4 つ以上のマルウェアソースコードに共通して含まれる）のプロセス名は 347 種類であった．この 347 種類のプロセス名に関連するソフトウェアをカテゴリごとに分類した結果を**表 1** に示す．表 1 に示すとおり，セキュリティソフトウェアは 191 件で全体の 55% を占めている．このうち，ウイルス対策に関するプロセスが 153 件，ファイアウォールに関するプロセスが 38 件であった．また，システムコマンドや自身以外のマルウェア（bbeagle.exe というプロセス名で動作する WORM_BAGLE.A など）も強制終了の対象となっていることが分かった．

4.　強制終了を逆用するマルウェアの活動抑止手法の提案

本章では，ソースコード解析の結果判明したプロセスの強制終了処理に着目したマルウェアの活動抑止手法を検討し，手法を実現するプロトタイプシステムの実装について述べる．

4.1　おとりプロセスを用いたマルウェア対策手法

調査結果では，マルウェアは強制終了するプロセスをプロセス名によって特定していること，複数のマルウェアが共通して強制終了の対象とするプロセス名が 347 種類存在することを示した．ソースコードが入手可能なマルウェアは，それを基にした亜種が大量に作成されることから，調査の結果得られた特徴が亜種間で引き継がれる可能性がある．そこで，マルウェアによる強制終了の対象となるプロセス名で実行されるおとりのプロセスを生成することによって，マルウェアを検知し，逆にその活動を抑止する手法を提案する．手順は次に示すとおりである．

(1)　おとりプロセスの生成
おとりプロセスは，マルウェアが強制終了の対象とするプロセス名で動作する．

(2)　おとりプロセスの監視とマルウェア判定
おとりプロセスが他プロセスから強制終了されることを検知した場合，強制終了を試みたプロセスをマルウェアプロセスと判定する．

(3)　マルウェアプロセスの強制終了
おとりプロセスの強制終了を阻止し，マルウェアと判定したプロセスを強制終了することで活動抑止を実現する．

4.2 実　　装

提案手法を実現するプロトタイプシステムについて述べる．まず，提案手法を実現するための要件と，実装したプロトタイプシステムの有効性を確認するための要件を示す．

- **提案手法を実現するための要件**
  - **＜要件 1＞**　強制終了を誘導するおとりプロセスを生成できること
  - **＜要件 2＞**　おとりプロセスの終了を検知できること
  - **＜要件 3＞**　強制終了と処理元プロセスを特定できること
  - **＜要件 4＞**　おとりプロセスは強制終了されないこと
  - **＜要件 5＞**　マルウェアと判定したプロセスを確実に停止できること
  - **＜要件 6＞**　複雑な処理を行わず高速に動作すること
  - **＜要件 7＞**　正規のプロセスによる強制終了をマルウェアとして誤検出しないこと
- **有効性検証のための要件**
  - **＜要件 8＞**　API の呼び出し記録をログとして保存できること

提案手法を実現するプロトタイプシステムの概要を**図 2** に示す．

4.2.1 おとりプロセスの生成と管理

マルウェアはプロセス名の照合によって強制終了の対象を決定しているという調査結果に基づき，次のように実装した．

おとりプロセスは，単一のテンプレートプログラムのファイル名のみが異なる複製を作成した後，これらの複製をウィンドウ非表示にして起動することで，複数のおとりプロセスとして動作させる（**＜要件 1＞**）．おとりプロセスを管理するおとり管理プログラムは，設定ファイルに列挙したおとりプロセス名を読み込み，テンプレートプログラムの複製および実行（プロセス生成）を行う．なお，プロトタイプシステムでは，ユーザモードで動作するアプリケーションとしておとり管理プログラムを実装した．

また，プロトタイプシステムには実装していないが，実用化にあたっての課題として，システム起動時におとり管理プロセスを自動起動すること，自動起動が無効化されないようファイルやレジストリの保護が必要である．

4.2.2 おとりプロセスの終了の検知

おとりプロセスの終了を検知するにあたっては，API フックを用いた．ただし，マルウェアには，ユーザモードにおける API フックを回避するものが存在するため，カーネルモードにおいて Windows のネイティブ API[9]（システムコール）をフックすることでシステム上で実行されるすべてのプロセスに対してフックを適用することとした．

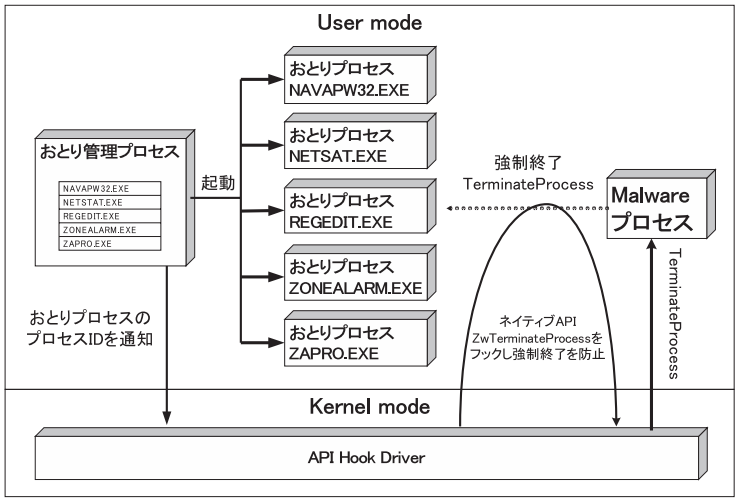

図 2　提案手法を実現するプロトタイプシステムの概要
Fig. 2　Outline of a prototype system.

実装の詳細は次のとおりである．プロセスの終了は，ユーザモードの Win32 API である ExitProcess API または TerminateProcess API の呼び出しによって実行される．その際，カーネルモードでは対応するネイティブ API として ZwTerminateProcess API が実行されることから，ZwTerminateProcess をフックすることにより，すべてのプロセスの終了を検知することが可能となる．

また，**＜要件 2＞**を満たすため，システムコールのアドレスが格納されている SSDT（System Service Descriptor Table）を書き換える SSDT Hooking[10] という手法を用いて，ZwTerminateProcess API の呼び出し時に代替処理を挿入するカーネルモードで動作するドライバとして実装した．

4.2.3 マルウェア判定方法

提案方式におけるマルウェア判定の方法について述べる．前述の ZwTerminateProcess API の第 1 引数には，終了するプロセスのプロセスハンドルが格納されている．

このプロセスハンドルからプロセスの基本的な情報が格納されるカーネル構造体 PROCESS_BASIC_INFORMATION を参照することで，終了するプロセスのプロセス ID（以降，TargetPID とする）を取得する．次に，PsGetCurrentProcessID API を呼び出し，ZwTerminateProcess API を呼び出したプロセスのプロセス ID（以降，CurrentPID とする）を取得する．

これらの 2 つの API を用いることで，終了処理を指示するプロセスと終了処理を指示されたプロセスの両方のプロセス ID を取得できることから，フックした ZwTerminateProcess API の代替処理として実装した．プロセスの正常な終了は，終了するプロセス自身による ExitProcess API の呼び出しによって行われ TargetPID と CurrentPID が一致する．一方，TargetPID と CurrentPID が一致しない場合は自身以外のプロセスからの終了指示であることから，次のようにしてマルウェア判定を行う（**<要件 3>**）．

(1) おとり管理プロセスは，おとりプロセスの生成後，ドライバ（API Hook Driver）に対しておとりプロセスとおとり管理プロセス自身のプロセス ID を通知する．

(2) ドライバ（API Hook Driver）には，任意のプロセスが終了される際に TargetPID が通知される．このとき事前に登録されているおとりプロセスのプロセス ID と比較を行い，一致した場合，おとりプロセスまたはおとり管理プロセスが他プロセスから強制終了されていると考え，CurrentPID のプロセス ID で実行されているプロセスをマルウェアと判断する．

4.2.4　マルウェアの強制終了

前述のマルウェア判定処理は，ZwTerminateProcess API をフックした際に行っている．マルウェアによるおとりプロセスの強制終了を検知した場合には，CurrentPID から取得したプロセスハンドルを引数として正規の ZwTerminateProcess API をただちに呼び出す．これにより，マルウェアのプロセスを強制終了させ，おとりプロセスの強制終了を回避することができる（**<要件 4>**，**<要件 5>**および**<要件 6>**）．

4.2.5　正規処理判定方法

Windows システム標準のタスクマネージャによる正規の手順でのプロセス強制終了は次に示す方法で実現した（**<要件 7>**）．

タスクマネージャはおとりプロセスのようにあらかじめ起動していないので，プロセス ID によって判定することができない．そのため，ZwQueryInformationProcess API によって ProcessImageFileName を取得した CurrentPID のプロセスの実行ファイルパスがタスクマネージャのパス（C:\WINDOWS\system32\taskmgr.exe）である場合は，正規の手順による強制終了と判定する．

4.2.6　API 呼び出しとマルウェア抑止処理の記録

プロトタイプシステムでは，提案手法が有効に機能しているかどうかを確認するために，マルウェアによるおとりプロセス強制終了の試みを記録する機能と，マルウェア自身のプロセス強制終了を記録する機能を実装した．実現にあたっては，ドライバ（API Hook Driver）に ZwTerminateProcess API の呼び出し時刻と引数，TargetPID と CurrentPID をログに出力する処理を実装した（**<要件 8>**）．

5. 評　　価

本章では，提案手法を実装したプロトタイプシステムの評価について述べる．

5.1　評 価 項 目

次に示す 2 つの評価項目について，ハニーポット[*1]で収集したマルウェア検体を用いて実施した．

なお，検体捕捉時点でウイルス対策ソフトウェアによって検出できたものを既知検体，検出不可であったものを未知検体とする．既知検体は検出名ごとに 1 種と数え，未知検体はファイルのハッシュ値ごとに 1 種とする．

(1) **プロトタイプシステムの動作検証**
実装したプロトタイプシステムが期待どおりに動作し，マルウェアのプロセスを停止できることを**表 2** に示す検体群 1 を用いて確認する．
- **検体群 1**
  2007 年 3 月から 2007 年 7 月の間にハニーポットで収集した 1,766 種類のマルウェア検体

(2) **著名なボットとトロイの木馬型のマルウェアに対する有効性の確認**
著名なボットとトロイの木馬型のマルウェアに対する有効性を**表 3** に示す検体群 2 を用いて確認する．
- **検体群 2**
  2007 年 4 月から 2008 年 1 月までにハニーポットで収集した検体 1,619 種類

5.2　評 価 手 順

プロトタイプシステムを導入した仮想環境を構築し，API 呼び出しとマルウェアの活動

⋆1 サイバークリーンセンターが運用するハニーポット

表 2　検体群 1
Table 2　Sample Group 1.

種類	数
既知検体で検出名に“ BOT ”を含む	1,115
上記以外の既知検体	175
未知検体	500
合計	1,766

表 3　検体群 2
Table 3　Sample Group 2.

種類	数
著名なボットファミリ	916
トロイの木馬（接頭辞 TROJ）	703
合計	1,619

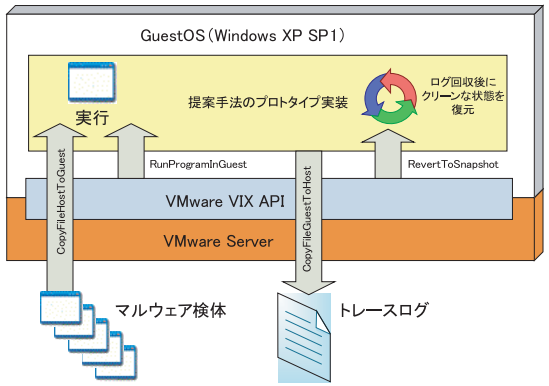

図 3　実験システム
Fig. 3　Evaluation environment.

抑止処理の記録機能を利用して，おとりプロセスの強制終了を試みたが，逆に強制終了されたマルウェア検体を集計する．なお，評価にあたっての前提条件として，おとりプロセスが使用するプロセス名は，調査結果において共通度 4 以上のプロセス名のうち，著名なウイルス対策ソフトウェアのプロセス名を中心とする 100 種類を使用した．

また，大量の検体を用いて評価を行うにあたり，VMware に対して，下記の一連の処理を繰り返し行う必要があるため，VMware VIX API を用いて自動化する実験システム（**図 3**）を構築し実施した．

（1）　評価環境へのマルウェア検体のコピー
（2）　評価環境内でのマルウェア検体の実行
（3）　評価環境からの API フックログのコピー
（4）　評価環境のスナップショット復元

実験システム内の環境は次のとおりである．

- GuestOS は Windows XP SP1（英語版）
- VMware の GuestOS 内であることを隠蔽するファイルやレジストリの変更は加えない
- 1 つのマルウェア検体を実行するごとに，VMware のスナップショットの復元機能を用いてクリーンな環境に戻す（検体の実行時間は 120 秒間）．
- ネットワークに接続する

5.3　結　　果

検体群 1 については，13 種のマルウェア検体が，おとりプロセスの強制終了を試み，さらに，プロトタイプシステムの動作によりマルウェア検体のプロセスの強制終了を確認した（**表 4**）．このことから，これら 13 種に対しては提案方式が有効であると判断できる．

また，検体群 2 については，表 4 に示す 6 種のマルウェア検体において，プロトタイプシステムの動作によりマルウェア検体のプロセスの強制終了を確認した．

5.4　考　　察

（1）　提案方式の有効性

評価の結果，本論文で提案するプロセス強制終了を利用したマルウェアの抑止手法が，著名なボットや近年増加傾向にあるトロイの木馬型のマルウェア[11] に対して有効であることを確認した．特に検体群 1 に含まれる WORM_RBOT.CWO は 2007 年 8 月度にサイバークリーンセンターにおいて 7,048 件捕捉されており，検出名に BOT を含むもの 1,115 種の中で 8 番目に多い種である[12]．このことから，実際のインターネットに多く蔓延しているボットに対しても有効であると考えられる．また，提案手法が有効であったマルウェア検体数は少ないものの，未知の検体に対して効果があったことは，既存の主要検知技術であるパターンマッチングを補完する技術として利用できることを示している．さらに，提案手法は，複数のウイルス対策ソフトウェアとして振る舞うおとりのプロセスを動作させ，検知するという点に特徴があ

表 4　有効性の確認できた検体と強制終了を試みられたおとりプロセス
Table 4　Terminated sample.

検体群 1	
検出名称または SHA-1	強制終了対象
WORM_RBOT.CHO	FIREWALL.EXE
WORM_RBOT.CWO	FIREWALL.EXE
WORM_RBOT.FPQ	Honeyprocess.exe
WORM_SDBOT.BSR	FIREWALL.EXE
WORM_SDBOT.DYX	FIREWALL.EXE
WORM_SDBOT.ELS	Honeyprocess.exe
WORM_SPYBOT.ABP	Honeyprocess.exe
WORM_SPYBOT.OS	FIREWALL.EXE
WORM_IRCBOT.ZO	Honeyprocess.exe
38baa0a574e78d1af36672ce8dd93145eea7fd4e	UPDATE.EXE
4c8eb028c4307a5764855c4b61211dc80283da64	UPDATE.EXE
a0d7e270b253056ca1f18e7fd9982421c59d7cb9	FIREWALL.EXE
a155907f26ae4ea3dccaa854aa01205d12fc6379	UPDATE.EXE
検体群 2	
検出名称	強制終了対象
WORM_RBOT.GEN	AVCONSOL.EXE
WORM_SDBOT.EZI	ALERTSVC.EXE
BKDR_RBOT.DY	AVCONSOL.EXE
TROJ_AGENT.FFY	UPDATE.EXE
TROJ_XPACK.BA	AVCONSOL.EXE
TROJ_UTOTI.AA	NAVW32.EXE

る．この特徴は，ウイルス対策ソフトウェアの振舞い検知機能を補強することができるため，既存のウイルス対策ソフトと協調したマルウェアの検知と被害抑制を実現できると考える．ただし，おとりプロセスを大量に生成すると OS 資源の消費によって体感速度の低下などのトレードオフが考えられる．実用化にあたっては，調査を通して，おとりプロセス名の絞り込みを行うなどを検討していきたいと考えている．

(2) 強制終了を試みられたおとりプロセス

強制終了を試みられたおとりプロセス名は，表 4 に示すとおりである．検体群 1 ならびに検体群 2 において結果に差があるが，これは，評価項目 1 と評価項目 2 でおとりプロセスの起動順序が異なっていたこと，マルウェアによる強制終了対象プロセスの探索アルゴリズムに違いある可能性が考えられる．なお，FIREWALL.EXE は評価項目 1 において最初に起動したおとりプロセスであり，AVCONSOL.EXE は評価項目 2 において 4 番目に起動したおとりプロセスである．いずれの場合も起動順の早いおとりプロセスで検知に成功している．

プロトタイプシステムでは，調査結果において共通度 4 以上のプロセス名のうち，著名なウイルス対策ソフトウェアのプロセス名を中心とする 100 種類を使用したが，効率的なおとりプロセスの生成と管理については，今後の課題である．

表 5　マルウェアの自己防衛機能およびその他の挙動
Table 5　other bihavior.

挙動	件数
CreateProcess でプロセスを生成	1,176
バッチファイル（.bat）を作成	758
hosts ファイルを改変	18
ドライバ（.sys）を生成	14
仮想マシン環境を検知して終了	121

(3) マルウェアの自己防衛機能およびその他の挙動

検体群 1 の評価において，API の呼び出し記録から確認できたマルウェアの自己防衛機能およびその他の挙動を**表 5** に示す．このうち，2 章の関連研究であげた自己防衛機能については，セキュリティ機能の無効化（対策情報の遮断）に該当する hosts ファイルの改変が 18 件，動的解析の妨害に該当する仮想環境の検知が 121 件であり，約 1 割に実装されていることを確認した．また，新たな知見として，API の呼び出しとともに記録されたデバッグ情報から，仮想環境の検知は，Themida[⋆1] や NTkrnl Packer[⋆2] などの商用 Packer の機能が利用されていることが分かった．これら商用 Packer は仮想環境の検知だけでなく難読化の機能も備えており，動的解析と静的解析の両方を妨害するものである．本来有益なソフトウェアの保護を目的としたセキュリティ技術が，マルウェアに悪用されている実態が明らかになった．今後もマルウェアでのセキュリティ技術の利用状況を調査していくとともに，悪用されにくいセキュリティ技術について検討していきたいと考えている．

6. おわりに

本論文では，マルウェアによるセキュリティ無効化機能の 1 つであるプロセスの強制終了攻撃を利用した対策手法として，おとりプロセスを用いた検知および強制終了手法を提案し

⋆1 Themida は，Oreans Technologies の登録商標または商標です．
⋆2 NTkrnl Packer は，NTkrnl Software, Inc. の登録商標または商標です．

た．また，提案手法を実現するプロトタイプシステムの実装を行い，ハニーポットで捕捉したマルウェア検体を用いた評価を通して，その有効性を確認した．

マルウェアのセキュリティ無効化攻撃は，多くのマルウェアに利用され，技術自体も進化していくと推測される．ウイルス対策ソフトウェアによる対策を強化する技術として，マルウェアの自己防衛機能を逆に利用することは有用であると考える．今後も，マルウェアの自己防衛機能の詳細を解析，それらを逆用する活動抑制手法の検討とともに，提案手法の実用化を目指していきたいと考えている．

謝辞　本研究の一部は Telecom-ISAC Japan，サイバークリーンセンターの支援を受け実施している．本研究を進めるにあたり，有益な助言と協力をいただいた Telecom-ISAC Japan の関係者各位に深く感謝いたします．また論文執筆にあたり，有意義なご指摘をいただいた田中優毅氏に感謝いたします．

参　考　文　献

1) 高橋正和，村上純一，須藤年章，平原伸昭，佐々木良一：フィールド調査によるボットネットの挙動解析，情報処理学会論文誌，Vol.47, No.8 (2006).
2) 株式会社フォティーンフォティ技術研究所：AADebug v0.12 (2007). http://www.fourteenforty.jp/research/freeware.htm
3) 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，情報処理学会シンポジウムシリーズ，Vol.2008, No.8 (2008).
4) 松木隆宏，村上純一：悪性プログラムの耐解析機能を逆用した活動抑制手法の提案，*Computer Security Symposium 2006* (*CSS2006*) (2006).
5) UPX: UPX: the Ultimate Packer for eXecutables. http://upx.sourceforge.net/
6) 岩村　誠，伊藤光恭，村岡洋一：コンパイラ出力コードモデルの尤度に基づくアンパッキング手法，情報処理学会シンポジウムシリーズ，Vol.2008, No.8 (2008).
7) Websense Security Labs: StormWorm & Botnet Analysis (2008). http://securitylabs.websense.com/content/Assets/Storm_Worm_Botnet_Analysis_-_June_2008.pdf
8) 松木隆宏，寺田真敏：セキュリティ無効化機能を逆用したマルウェア活動抑制手法の検討，*Computer Security Symposium 2007* (*CSS2007*) (2007).
9) The Metasploit Project: Windows System Call Table (NT/2000/XP/2003/Vista). http://www.metasploit.com/users/opcode/syscalls.html
10) Hoglund, G. and Butler, J.: *Rootkits: Subverting the Windows Kernel*.
11) マイクロソフト：マイクロソフト セキュリティ インテリジェンス レポート（2007 年 7 月–12 月）．http://www.microsoft.com/downloads/details.aspx?displaylang=ja&FamilyID=bcc879db-9fe6-4331-b231-e274ea8fc804
12) サイバークリーンセンター：2007 年 8 月度サイバークリーンセンター活動実績 (2007). https://www.ccc.go.jp/report/200708/0708monthly.html

(平成 20 年 12 月 1 日受付)
(平成 21 年　6 月 4 日採録)

松木　隆宏（正会員）

（株）ラック サイバーリスク総合研究所研究員．2005 年岡山大学工学部通信ネットワーク工学科卒業．同年（株）ラック入社．ネットワークセキュリティ脅威分析等のセキュリティコンサルティング部門を経て，マルウェアの調査研究，マルウェア対策技術の研究開発に従事．2006 年より，サイバークリーンセンターによるボット対策プロジェクトに参画し，調査研究，ハニーポットの開発，運用支援に従事．2007 年から安心・安全インターネット推進協議会 P2P 研究会構成員．2008 年情報処理学会コンピュータセキュリティ研究会専門委員．CISSP．

新井　　悠

（株）ラック サイバーリスク総合研究所所長．2000 年（株）ラック入社．コンピュータセキュリティ研究所にて脆弱性の分析，R&D 部門の統括，ネットワークセキュリティ脅威分析等のコンサルティング業務やセキュリティ・アドバイザを経て，現職．著書・監修書として『ネットワーク攻撃詳解—攻撃のメカニズムから理解するセキュリティ対策』（ソフト・リサーチ・センター),『インシデントレスポンス』（翔泳社）等．2004 年総務省「次世代 IP インフラ研究会」セキュリティWG 構成員，2005 年内閣官房 NIRT（緊急対応支援チーム）研修講師，2006 年経済産業省「ウェブアプリケーションセキュリティガイドライン策定 WG」委員，2007 年総務省次世代の情報セキュリティ政策に関する研究会」構成員を務める．CISSP．

寺田　真敏（正会員）

（株）日立製作所システム開発研究所主管研究員．1986 年（株）日立製作所入社．システム開発研究所にてネットワークセキュリティの研究に従事．博士（工学）．2002 年から 2006 年慶應義塾大学大学院社会人学生として，JPCERT/CC Vendor Status Notes プロジェクトに参画．2004 年 4 月から JPCERT コーディネーションセンター専門委員，中央大学研究開発機構客員研究員，2004 年 8 月から（独）情報処理推進機構セキュリティセンター研究員，2004 年 10 月から（株）日立製作所　Hitachi Incident Response Team チーフコーディネーションデザイナを兼務．著書に,『基礎からわかる TCP/IP セキュリティ実験』オーム社 2000 年等.

土居　範久（正会員）

1969 年慶應義塾大学大学院博士課程単位取得退学．慶應義塾大学理工学部教授を経て，2003 年より中央大学理工学部教授，慶應義塾大学名誉教授．工学博士．現在，文部科学省科学技術・学術審議会委員，総務省情報通信審議会委員，世界科学会議（International Council for Science（ICSU））Priority Area Assessment Panel of Scientific Data and Information メンバ，科学技術振興機構（JST）社会技術システムミッションプログラム I 「I 情報セキュリティ」研究統括，特定非営利活動法人日本セキュリティ監査協会会長，国際計算機学会（ACM）日本支部長，等．専門はソフトウェアを中心とした計算機科学.