おとりの文書を表示する標的型マルウェア検知手法の評価

高橋 佑典† 渡部 正文†‡ 島 成佳†‡ 吉岡 克成‡

†日本電気株式会社
211-8666 川崎市中原区下沼部 1753
{y-takahashi@mb, watanabe@cj, shima@ap}.jp.nec.com

‡横浜国立大学大学院環境情報研究院/先端科学高等研究院
240-8501 横浜市保土ケ谷区常盤台 79-7
yoshioka@ynu.ac.jp

あらまし 攻撃者は様々な手段，謀略を講じてマルウェアに感染させようとする．標的型攻撃ではその手口が巧妙で，正規の業務メールとの見分けが困難なメールに，マルウェアを添付して組織内のユーザに送信する．攻撃者はこのマルウェアを，入口対策をすり抜けるように作成するため，組織内のユーザへ届いてしまう．添付したマルウェアを会議資料などの文書ファイルと偽ることで，ユーザがマルウェアを実行するよう誘導し，マルウェア実行後におとりの文書を表示してマルウェア感染をカモフラージュする．このような攻撃者の巧みな手口により，標的型攻撃の被害者はマルウェア感染を完全に避けることは困難である．標的型攻撃による情報漏洩を未然に防ぐためには，攻撃者に情報を奪取される前に攻撃を検知する必要がある．そこで，我々はマルウェア感染直後のプロセス構造に着目し，おとり文書を表示するマルウェアの検知手法を提案した．本稿では，新しいデータセットを用いてエンドポイント型の製品と比較実験を行ない，提案手法の有効性を考察する．

Evaluation of Detection Method of Targeted Malware Displaying a Decoy Document

Yusuke Takahashi† Masafumi Watanabe†‡ Shigeyoshi Shima†‡ Katsunari Yoshioka‡

†NEC Corporation
1753 Shimonumabe, Nakahara-Ku, Kawasaki 211-8666, JAPAN
{y-takahashi@mb, watanabe@cj, shima@ap}.jp.nec.com
‡Graduate School of Environment and Information Sciences / Institute of Advanced Sciences,
Yokohama National University
79-7 Tokiwadai, Hodogaya-Ku, Yokohama 240-8501, JAPAN
yoshioka@ynu.ac.jp

Abstract Attackers compromise target hosts by various methods and tricks. In targeted attacks, especially, the tricks are sophisticated. For example, attackers send users in the organization the e-mails which look like regular business mails and include the malware attachments that slip through the entry control measures. Attached malware entice users to install themselves by pretending to be business documents. Due to tricks like this, it is difficult for victims to avoid malware infection. Therefore, for purpose of preventing information leakage, we have to detect targeted attacks before attackers steal information. We have proposed a method to detect the malware that display decoy document by focusing on process tree created immediately after compromises. In this paper, we experiment with new dataset, and show the effectiveness of our method by comparison with commercial endpoint appliances.

1　はじめに

企業や官公庁の保有する顧客情報や機密情報の奪取を目的とした標的型攻撃が年々脅威を増している[1, 2]. 2015年は，標的型攻撃による情報奪取事件が次々と明るみに出ており，第2四半期だけで15の組織が被害を受けたことが発覚している[3]. 特に，公的機関の保有する個人情報が大量に奪取された事件は社会に不安を与えた[4].

個人情報や顧客情報が流出し悪用されると，非常に多くの人が詐欺や架空請求などの犯罪の被害に遭う可能性が高まる. 企業は，顧客や取引先，株主，従業員を含むステークホルダーによって支えられているため，ひとたび情報漏洩が起こると，ステークホルダーからの信用を失い，業務中断や取引停止などによって，存続が危ぶまれる. そのため，情報漏洩を未然にくい止めることが非常に重要である.

多くの組織は，標的型攻撃への対策として組織内への侵入を防ぐ入口対策を実施している[5]. しかし，攻撃者は入口対策をすり抜けるマルウェアをメールに添付し，組織内のユーザに送信する. 攻撃者は，メール本文に実在の人物の署名や業務に関連のある内容を記述するなどして，受信者に業務に関するメールであると錯覚させる. また，添付したマルウェアを製品やサービスに関する問い合わせのような業務に関連のある文書ファイルと騙ることで，受信者に添付ファイルを開かせてマルウェア感染するよう誘導する. このようなメールは標的型攻撃メールと呼ばれており[6]，受信者が標的型攻撃メールと正規のメールを区別することは非常に困難である. 近年では，メール本文に合致するおとりの文書ファイルを表示することで，感染後も受信者に気付かれないようにするマルウェアが確認されている[7, 8].

このような状況から，マルウェア感染を前提とした標的型攻撃対策が重要となっている. 本研究では，第2章で述べるサイバー・キル・チェーン[9]の7つの段階のうち，インストール段階から目的実行（情報奪取）段階までの間に標的型攻撃を早期に検知し，情報漏洩を未然に防ぐことを目指している. そこで，筆者らは，おとり文書を表示するマルウェアの起動直後の挙動に着目した標的型マルウェア検知手法を提案した[10]. 論文[10]では，おとり文書を表示する標的型マルウェア12検体に対して実験を行ない，12検体すべてを検知できることを示した. しかし，論文[10]では Process Monitor[11]のログを用いた事後的な提案手法の適用であり，他の製品や手法との比較を行なっていなかった. 本稿では，検知アルゴリズムを改良し，ホストの内部挙動監視機能を持った試験プログラムを実装する. そして，試験プログラムのマルウェア検知率を 2 つの製品と比較し，提案手法の有効性を考察する.

2　本研究の課題

攻撃者の活動は，図 1のように，偵察，武器化，配送，攻撃，インストール，遠隔操作の段階を経て，最終的な目的実行（情報盗取）に至る. これら一連の活動のつながりは，サイバー・キル・チェーンと呼ばれる[9]. 情報漏洩を未然に防ぐために，標的となった組織は，攻撃者の活動が目的実行（情報盗取）段階に至る前に，このサイバー・キル・チェーンを断ち切らなければならない.

標的型攻撃メールによる攻撃の場合，偵察ステップにおいて，攻撃者は標的組織や標的組織内の人物の情報収集を行なう. 標的となる組織がこの偵察活動を防ぐことはできない. また，武器化活動についても，武器化は攻撃者側で行われるため，標的となる組織が防ぐことはできない. 攻撃者は，組織が標的型攻撃メールの対策として実施している入口対策に検知されないマルウェアを作成することができる. シグネチャベースのセキュリティアプライアンスに対しては，シグネチャに定義されていないマルウェアを作成することで検知をすり抜けることができる. サンドボックス上でマルウェアの振る舞いを検知するアプライアンスに対しても，仮想環境を検出することで検知を回避するマルウェアを作成できる[12]. 攻撃者は，このようなマルウェアを正規の業務メールに見せかけたメールに添付し，送信することで，組織内のユーザにマルウェアを実行させる. そのため，組織は配送や攻撃の活動を完全に防ぐことも困難である. 目的実行（情報盗取）段階において攻撃者に重要情報を持ちだされないように，C&C サーバや悪性ページを検知する出口対策を実施する組織も多い. しかし，正規のサーバが攻撃者に乗っ取られている場合があるため，これも完全に防ぐことは困難である[13].

情報漏洩のリスクを最小化するために，組織は，入口対策・内部対策・出口対策を網羅的に実施する必要がある. 内部対策は，攻撃者が組織内部で行なうインストール，遠隔操作の活動を検知し，これらを止めることを目的とした対策である.

組織内部における攻撃活動の発見が早いほど，メリットがある. 特に，攻撃，もしくは，インストールの段階で攻撃者による活動を検知し止めることができれば，感染が組織内

部の他端末に広げることもなく，その後の端末における駆除作業などにかかるコストを抑えることができる．

これより，リスクとコストの最小化のため，本研究の目的をインストール活動時に標的型攻撃を早期に検知することとする．本稿では，おとり文書を表示して受信者の不信感を回避するマルウェアを，インストール活動時に早期検知することを目指す．

図 1 サイバー・キル・チェーン

3 課題解決のアプローチ

2章で述べたように，本論文で検知対象としているのはマルウェアのインストール活動時の挙動であるため，着目するのは端末内部の挙動である．

受信者が標的型攻撃メールに添付されたマルウェアを実行してしまうアプリケーションは，以下の 3 種類である．

① メールクライアント：
「添付ファイルを開く」のような機能を用いて実行

② ブラウザ：
Web メールを使用している場合，添付ファイルをダウンロードした際に現れるダイアログから実行

③ ファイルエクスプローラ：
一旦ローカルの任意のディレクトリに保存した後，ファイルエクスプローラでそのディレクトリを表示し，保存したファイルを実行

マルウェアのプロセス（以降，マルウェア・ルートプロセス）は，上記 3 種類のアプリケーションのプロセスから生成される．3 種類のアプリケーションのプロセスを，マルウェア感染の起点となるプロセスとし，起点プロセスと呼ぶこととする．

おとり文書表示マルウェアは，ファイルタイプの観点で大きく2種類に分けられる．「文書ファイルを装った実行可能ファイル」と「文書閲覧アプリケーションの脆弱性を突く悪性文書ファイル」である．悪性文書ファイルは不正活動を行なうために，アプリケーションの脆弱性を突く必要があるが，実行可能ファイルはその必要がなく，実行後すぐに不正活動を始めることができる．このような実行後の挙動の違いはあるが，おとり文書表示マルウェアに共通する特徴は，正規の文書閲覧アプリケーションを用いておとりの文書を表示することである．

おとり文書の表示は，マルウェア・ルートプロセスから生成される文書閲覧アプリケーションのプロセスで行われる．このとき，マルウェアは，攻撃者がその後の活動に使用するためのバックドアを作成する．バックドアの作成は，マルウェア・ルートプロセス内，もしくは新たに生成するバックドア作成用のプロセスで行われる．

以上より，マルウェア・ルートプロセスは，起点プロセスの子孫であり，文書閲覧アプリケーションの祖先であるといえる．本稿では，バックドアの作成や C&C サーバへの通信などの挙動は考慮せず，このプロセス構造のみに着目してマルウェアのインストール活動を検知する．プロセス構造のリアルタイム把握，という観点から，検知には以下の機能が必要である．

・プロセス生成の様子を監視する機能

・なりすましプロセスと正規プロセスを識別する機能

ホストのプロセス生成の様子を監視し，マルウェア・ルートプロセスを特定する試験プログラムを実装する．

4 実装

ホストの内部挙動を監視し，観測した挙動からマルウェア・ルートプロセスを特定する試験プログラムを作成した．その際，論文[10]で提案した検知アルゴリズムの改良を行なった．

4.1 検知アルゴリズムの改良

これまで観測してきたおとり文書を開く悪性文書ファイルは，マルウェア・ルートプロセスが，不正活動を行なうプロセスとおとり文書を表示するプロセスを分岐させていた．これを受けて，論文[10]では悪性文書ファイルの検知アルゴリズムとして，「プロセス p の子孫にあたるプロセスに，子プロセスを 2 つ以上生成するプロセスが存在すること」と提案していた．しかし，図 2のように，マルウェア・ルートプロセスから生成された不正活動プロセスがおとり文書表示プロセスを生成することも可能であり，その場合でも検知できるように，以下のように悪性文書ファイルの検知アルゴリズムを改良した．

検査対象のプロセスを p，起点プロセスの集合を R，文書閲覧アプリケーションのプロセスの集合を D としたとき，$R \cap D = \varnothing$ であり，検知アルゴリズムは以下のように表せる．

子プロセスを分岐する場合：

・$p \in D$

・p の子孫にあたるプロセスに，子プロセスを 2 つ以上生成するプロセスが存在する

・p の子孫にあたるプロセスに，p と同じアプリケーション

のプロセス（おとり文書を表示するプロセス）が存在する

子プロセスを分岐しない場合:

・p∈D

・p の子孫にあたるプロセスに，子プロセスを 2 つ以上生成するプロセスが存在しない

・p の孫以下のプロセスに，p と同じアプリケーションのプロセス（おとり文書を表示するプロセス）が存在する

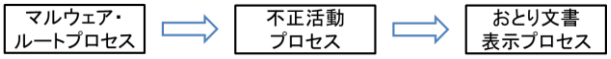

図 2 おとり文書表示マルウェアのプロセス

4.2 試験プログラムの概要

試験プログラムは，起動時に実行中のプロセス一覧を取得し，その後新たに生成されるプロセスを監視する．新たに生成されたプロセスが文書閲覧アプリケーションのものであった場合，検知フェーズに移行する．以下に，検知フェーズの動作を記述する．

1. 観測した文書閲覧アプリケーションdを保持
   docapp = d, , p = d, depth(検査深度) = 0, FLAG = False,
   mrp(マルウェア・ルートプロセス) = None
2. pの親プロセスp′の情報を取得．depth += 1
3. p′∈Rのとき，検知フェーズを終了する
   このとき，mrp != Noneならば検知アラートを表示する
4. p′が子プロセスを2つ以上生成している場合，FLAG = True
5. 以下のいずれかの条件を満たすとき，mrp = p′
   – p′∉R, D
   – p′∈D，かつ，p′ == docapp，かつ，FLAG == True
   – p′∈D，かつ，p′ == docapp，かつ，depth > 1
6. p = p′とし，手順2以降を繰り返す

試験プログラムは Python3.4[14]で実装した．プロセス生成の監視機能は，WMI[15]を利用することで実現した．また，実装の際，予め実験環境に合わせて文書閲覧アプリケーションと，マルウェア感染の起点となるアプリケーションを設定した．マルウェアが文書閲覧アプリケーションのプロセス，もしくは，起点プロセスになりすますプロセスを生成していた場合，当プロセスをなりすましプロセスと判断するため，各アプリケーションのアプリケーション名とアプリケーションパスの対応も設定した．設定したアプリケーション名とアプリケーションパスを表 7に示す．この対応が一致している場合，正規のアプリケーションによるプロセスと判断する．

5 評価実験

提案手法を評価するため，おとり文書を表示する検体セットを用いて，振る舞い検知を行なう既存の製品と，作成した試験プログラムのマルウェア検知率を比較した．実験に使用した検体セットの構成を表 1に示す．検体セット中の各検体を＜ファイルエクスプローラ，メーラ，ブラウザ＞から実行し，提案手法と既存製品で検知できるか調査し，検知率を算出した．

表 1 検体セットの構成

		おとり文書のファイルタイプ				
		doc	xls	jsd	pdf	**計**
検体のファイルタイプ	exe	17	1	0	7	**25**
	doc	4	0	0	0	**4**
	xls	0	4	0	0	**4**
	jsd	0	0	1	0	**1**
	計	**21**	**5**	**1**	**7**	**34**

5.1 概要

本実験では，ビヘイビアベースの検知について評価する．

試験プログラムは，マルウェア・ルートプロセスを正しく特定できている場合を「検知」とした．

製品 A は，シグネチャマッチング，振る舞い検知，レピュテーションによる検知を組み合わせた製品であり，シグネチャベースで検知できなかった検体に対して，振る舞い検知，レピュテーションによる検知を行なう．

製品 B はマルウェアのシグネチャを使用しないセキュリティアプライアンスである．まず，ローカルのシステムに保存された検体に対して静的解析を行ない，マルウェアと判定された場合，アラートを表示する．この静的解析は，検体実行後に生成された新たなファイルに対しても行われる．実行後の挙動を複数のビヘイビアベースの検知エンジンで解析し，疑わしい挙動を検知した場合，該当ファイルの削除や，処理の遮断を行なう．

製品 A, B では，ビヘイビアベースでの検知，という観点で評価するため，検体実行後の検知結果に着目する．また，検体実行後の検知がシグネチャとのパターンマッチング，静的解析によるものであった場合，該当検体は検知率算出の際に除外する．

3 種類のアプリケーションからマルウェアを実行する手順を以下に述べる．このような手順で意図的にインストール活動をスタートさせ，既存製品や試験プログラムがこのインストール活動を検知するかどうかを確認する．

・ファイルエクスプローラ:

任意のディレクトリにマルウェアをダウンロードし，ファイルエクスプローラ上でダブルクリックすることで実行

・ブラウザ:

Web サーバにアクセスし，任意のディレクトリにダウンロード．ブラウザのダウンロード完了を知らせるダイアログから実行

・メーラ:

メールに添付されたマルウェアを任意のディレクトリにダウンロード．メーラのダウンロード履歴から実行

検体セットは，実際の標的型攻撃メールに添付された検体と，VirusTotal[16]から入手した検体で構成されている．全 34 検体中，25 検体が実行可能ファイル（検体のファイルタイプが exe），9 検体が悪性文書ファイル（検体のファイルタイプが doc, xls, jsd）である．検体を事前に解析環境で実行し，おとり文書が表示されることを確認した．実行可能ファイルがおとり文書で利用するファイルタイプは doc が最も多く，次いで pdf だった．悪性文書ファイルは，検体のファイルタイプと同じ形式のおとり文書を利用していた．

5.2 実験環境

図 3に示す仮想環境を構築した．評価には，マルウェアが実行に成功する必要があるため，意図的に脆弱な環境を用意した．

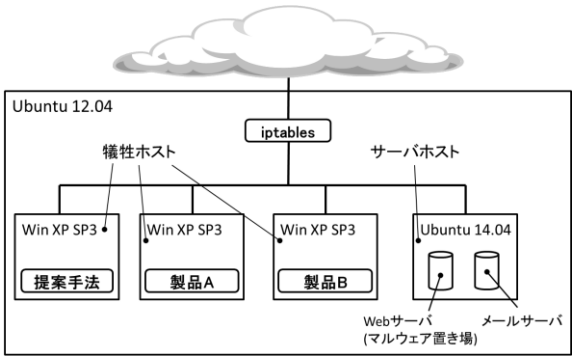

図 3 実験環境の概要

表 2 実験に使用したアプリケーション

種類	マシン	名称
ファイルエクスプローラ	犠牲ホスト	エクスプローラー
ブラウザ	犠牲ホスト	Internet Explorer 6
メーラ	犠牲ホスト	Thunderbird 38.1.0
文書閲覧アプリケーション	犠牲ホスト	Microsoft Office 2003 (Word, Excel)
		Adobe Reader 8.0
		三四郎 2010
Web サーバ	サーバホスト	SimpleHTTPServer (Python モジュール)
メールサーバ	サーバホスト	postfix 2.11.0 dovecot 2.2.9

Web サーバとメールサーバはサーバホスト (Ubuntu14.04) 上に構築し，マルウェアの実行は犠牲ホスト (Windows XP SP3) 上で行なった．また，メールサーバには予めマルウェアを添付したメールを保存した．

実験に使用したアプリケーションを表 2に示す．

5.3 実験結果

試験プログラムは全検体を検知することができたが，2 検体は5.1節に記述したマルウェア実行方法のうち 1 種類のみでしか検知できていなかった．

製品 A は，表 3のような検知結果となった．製品 A はマルウェアを実験環境に保存した時点で 14 検体を検知し，削除した．その場合は，削除されたファイルを検知アラート画面からリストアし，実行した．

表 3 製品 A の検知結果 内訳

		検体実行後				
		パターンマッチング	振る舞い		レピュテーション	検知せず
		検知・遮断	検知・遮断	検知（遮断せず）	検知（遮断せず）	
ローカルに置いた時点	検知	4	6	0	1	3
	検知せず	14	5	1	0	0

表 4 製品 B の検知結果（検出感度：既定）

		検体実行後		
		静的解析による検知	振る舞い検知	検知せず
ローカルに置いた時点	検知	0	0	17
	検知せず	0	12	5

表 5 製品 B の検知結果（検出感度：最高）

		検体実行後		
		静的解析による検知	振る舞い検知	検知せず
ローカルに置いた時点	検知	8	1	9
	検知せず	2	11	3

マルウェア実行後に表示された検知アラートに記された検知名を製品 A のホームページで調べ，＜シグネチャとのパターンマッチング，振る舞い検知，レピュテーションによる検知＞の 3 種類に分類した． 製品 B の検知結果を表 4，5に示す．製品 B はマルウェアを実験環境に保存した時点の静的解析により，既定レベルで 17 検体，最高レベルで 18 検体を検知した．これらはすべて実行可能ファイル形式の検体だった．実行前に検知された検体は，例外ファイルとして製品 B に設定し実行することで，検体実行後の挙動や生成されるファイルが検知されるか調べた．検体実行後の検知の種類は静的解析と振る舞い検知の 2 種類だった．

表 6に試験プログラム，既存製品の検知率を示す．表中の検知率は，[ビヘイビアベースで検知できた検体数]／[全検体数]で算出した．ただし，製品 A の検知率は，検体実行後にパターンマッチングで検知された 18 検体を除いた，16 検体 (表 3中の背景に色がついた検体)に対してのビヘイビアベースの検知率である． 同様に，製品 B の検知率も，静的解析によって検知された検体を除いたものである．(表 4，5の背景に色がついた検体)

5.3.1項以降に，結果の詳細を評価対象ごとに記述する．

表 6 マルウェア検知率

試験プログラム	製品 A※2	製品 B※2
100.0% (96.1%)※1	75.0%	35.3% (50.0%)※3

※1:[検知できた数]／[3 種類の実行方法＊全検体数]

※2:検体実行後にパターンマッチング，静的解析で検知された検体を除いている

※3:括弧内は検出感度が最高レベルの場合の検知率

5.3.1 試験プログラム

試験プログラムは，実験に使用した34検体すべてを検知できた．しかし，3 種類の検体実行手順のうち，ファイルエクスプローラから実行した場合のみ検知できた検体と，メーラから実行した場合のみ検知できた検体が，それぞれ 1 検体ずつあった．原因を調査したところ，マルウェアが生成するプロセスの 1 つを取得できておらず，検知フェーズでマルウェア・ルートプロセスまで辿れていないからであるとわかった．取得できなかったプロセスは，極めて短い時間しか生存していなかった．図 4の枠で囲われたプロセスが，試験プログラムで取得できなかったプロセスの 1 つである．該当プロセスは悪性文書ファイルが，おとり文書を表示するプロセスを起動するために生成したプロセスであると推測できる．おとり文書を表示するプロセスが生成された直後，該当プロセスは終了した．

プロセス取得漏れの原因調査のため，上記 2 検体を複数回実行した．その結果，検知できる場合とできない場合，すなわち，プロセス取得漏れが起こらない場合と起こる場合があった．また，Process Monitor でプロセス構造を確認していたが，いずれの場合も上記の短時間しか生存しないプロセスを含めて漏れ無く取得されており，プロセスの階層構造も同一だった．これらより，プロセス取得漏れの原因は観測機構の精度によるものと推測される．

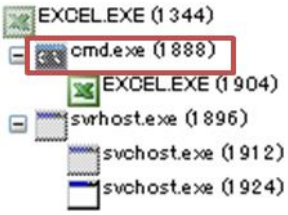

図 4 取得できなかったプロセス

5.3.2 製品 A

製品 A の振る舞い検知エンジンは，12 検体を検知でき，4 検体を検知できなかった．検知できた 12 検体のうち，11 検体については，対象ファイルの実行を未然に防いでいたが，1 検体はおとり文書が表示されてから，およそ 2 分後に検知アラートが表示された．ファイル実行の有無はマルウェアを実行する際に起動していた Process Monitor で確認した．

5.3.3 製品 B

製品 B の振る舞い検知エンジンは，検出感度が既定レベル，最高レベル共に 12 検体を検知した．検知された検体は，既定レベル，最高レベルで同一だった．検体セットに含まれる悪性文書ファイル 9 検体すべてについて，文書閲覧アプリケーションの脆弱性への攻撃を検知し，その後の処理を遮断していた．検知された実行可能ファイル 3 検体のうち，2 検体を新たに生成されたファイルが実行される前に検知し，遮断していたが，残りの 1 検体については，実行後の挙動を検知していた．これらは，製品 A と同様に，Process Monitor で確認した．

また，製品 B の振る舞い検知エンジンは，検出感度が既定レベルで 22 検体，最高レベルで 12 検体を検知できなかった．

6 考察

6.1 マルウェア検知

試験プログラムは，検体セットに含まれるすべてのおとり文書を表示するマルウェアを検知できた．本手法で着目している特徴は，おとり文書を表示するマルウェアに見られ，検知に有効であることがわかった．プロセスの生成のみを監視し，その構造に着目する，という単純なアルゴリズムで非常に高い検知率を得ることができた．5.3.1項で述べた通り，プロセス取得漏れの問題はプロセス観測機構の精度が原因と推測される．より精度の高いプロセス観測機構を用いることで，プロセス取得漏れの問題は解消すると考えられる．

一方で，提案手法はおとり文書を表示するマルウェアを対象としているため，おとり文書を表示しないマルウェアを検知できない．標的型攻撃メールでは，おとり文書を表示しないマルウェアや，メール本文に記述した攻撃サイトのURL リンクを開かせてドライブ・バイ・ダウンロードを試みる攻撃がある．標的型攻撃の早期検知の観点から，これらについても対応する必要があると認識している．

製品 A の振る舞い検知エンジンは，4 検体を見逃すという結果だった．評価基準をビヘイビアベースでの検知に設定したためこのような結果となったが，5.1節で述べたとおり，製品 A は複数の検知エンジンを持ったセキュリティアプライアンスである．検知エンジンすべてを統合した検知結果は，100%であった．しかし，2 章で述べたように，攻撃者はシグネチャに定義されていないマルウェアを作成し，組織内のユーザに送信することができる．仮に，見逃した 4 検体と挙動が似ており，かつ，シグネチャに定義されていないおとり文書表示マルウェアを攻撃者が送信してきた場合，製品 A だけではそのマルウェアを検知できない可能性がある．しかしながら，製品 A と本提案手法を併用することで，シグネチャに定義されていないおとり文書表示マルウェアを検知できる可能性が高まる．

製品 B についても，静的解析エンジンによる検知を除外したため，本来の検知率よりも低い結果となっている．静的解析エンジンによる検知を含めた検知率は，検出感度が既定レベルで 85.3%，最高レベルで 91.2%であった．すなわち，既定レベルで 5 検体，最高レベルで 3 検体を見逃している．製品 B はボットやバンキングマルウェアなど，他の脅威に対しても有効な製品である．本提案手法と併用することで，他の脅威に対しての防御を行いつつ，おとり文書表示マルウェアの検知精度を上げることができる．

今回，できるだけ多くの検体が動作するように Windows XP や Office 2003 を用意したが，実環境での有効性を考慮するには，より実状に合わせた環境での評価が必要になる．

6.2 端末への負荷と検知の所要時間

今回は評価を行なっていないが，提案手法は処理が軽く，既存製品と組み合わせて使用する場合に端末への負荷が小さいと考えられる．このように考える理由として，端末の挙動で監視するのはプロセス生成のみであり，ファイルアクセスや通信を考慮する必要がないことを挙げる．また，本提案手法は，導入環境に合わせて「起点プロセスとなりうるアプリケーション」と「文書閲覧アプリケーション」を設定する必要がある．端末のアプリケーションを統一している組織では，設定に掛かるコストが小さいと考えられる．

続いて，検知の所要時間について述べる．提案手法はおとり文書表示プロセスをトリガーに検知を行なうため，検知速度はマルウェアが実行してからおとり文書表示プロセスが生成されるまでの時間に依存する．Process Monitor で観測したプロセス起動時の時間から，マルウェア・ルートプロセス起動からおとり文書表示プロセス起動までの時間を算出したところ，3.3±2.8 秒掛かっていた．この時間は，端末の性能や CPU 使用率，メモリ使用率によって変動すると考えられる．おとり文書表示プロセスを観測してからマルウェア・ルートプロセスの特定までの時間は，実装方法によって変動する．今回はスクリプト言語である Python で実装したため，おとり文書表示プロセス観測からマルウェア・ルートプロセス特定まで，およそ 30 ミリ秒掛かっていた．

7 おわりに

本稿では，論文[10]で提案したおとり文書表示マルウェア検知手法について，リアルタイムにプロセスを監視し，おとり文書表示マルウェアのプロセスを検知する必要があったため，リアルタイムなプロセス監視機能を持つ試作プログラムを実装した．その際，論文[10]の検知アルゴリズムで見逃してしまうマルウェアをカバーするため，検知アルゴリズムの改良も行なった．既存製品の振る舞い検知方式とのマルウェア検知率を比較した実験では，提案手法は通信やファイルアクセスを考慮せず，プロセス構造のみに着目するという非常に単純なアルゴリズムであるにも関わらず，既存製品と同等の精度でおとり文書表示マルウェアを検知できることを示した．

今後は，攻撃の特徴を捉えたマルウェア検知，という観点から，おとり文書を表示しないマルウェアやドライブ・バイ・ダウンロードを利用する標的型攻撃の早期検知へと，提案手法の適用範囲を広げていく予定である．

参考文献

[1] “平成 26 年中のサイバー空間をめぐる脅威の情勢について,” 警察庁, [Online]. Available: http://www.npa.go.jp/kanbou/cybersecurity/H26_jousei.pdf. [Accessed: 11-Aug-2015].

[2] Symantec, “Internet Security Threat Report 2015 | Symantec,” 2015.

[3] TrendMicro, “2015 年第 2 四半期 セキュリティラウンドアップ,” 2015.

[4] “日本年金機構：個人情報125万件流出 ウイルスメールで - 毎日新聞,” *mainichi.jp*, 01-Jun-2015. [Online]. Available: http://mainichi.jp/select/news/20150602k0000m040001000c.html. [Accessed: 11-Aug-2015].

[5] TrendMicro, “組織におけるセキュリティ対策 実態調査 2015 年版,” 2015.

[6] IPA 独立行政法人 情報処理推進機構, “標的型攻撃／新しいタイプの攻撃の実態と対策,” 2011.

[7] “健康保険組合になりすました不審なメールについて,” セキュリティ研究センターブログ, [Online]. Available: http://blog.macnica.net/blog/2014/11/post-fca5.html. [Accessed: 11-Aug-2015].

[8] “医療費通知に偽装した攻撃（Backdoor.Emdivi）その後,” セキュリティ研究センターブログ, [Online]. Available: http://blog.macnica.net/blog/2015/01/post-39d4.html. [Accessed: 11-Aug-2015].

[9] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” *Lead. Issues Inf. Warf. Secur. Res.*, vol. 1, p. 80, 2011.

[10] 髙橋 佑典, 吉川 亮太, 吉岡 克成, 松本 勉, “ダミー文書表示に着目した標的型マルウェア検知手法,” *電子情報通信学会技術研究報告 IEICE Tech. Rep. 信学技報*, vol. 114, no. 489, pp. 157–161, 2015.

[11] Process Monitor, https://technet.microsoft.com/ja-jp/sysinternals/bb896645.aspx.

[12] S. Hardy, M. Crete-Nishihata, K. Kleemola, A. Senft, B. Sonne, G. Wiseman, P. Gill, and R. J. Deibert, “Targeted threat index: Characterizing and quantifying politically-motivated targeted malware,” in *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[13] TrendMicro, “標的型サイバー攻撃分析レポート 2015 年版 ～「気付けない攻撃」の高度化が進む～,” 2015.

[14] Python, https://www.python.org/.

[15] “Windows Management Instrumentation (WMI) の概要.” [Online]. Available: https://msdn.microsoft.com/ja-jp/library/Cc736575(v=WS.10).aspx. [Accessed: 11-Aug-2015].

[16] VirusTotal, https://www.virustotal.com/.

表 7 試験プログラムの設定

設定項目	アプリケーション名	アプリケーションパス
エクスプローラー	Explorer.EXE	C:¥WINDOWS¥Explorer.EXE
Internet Explorer 6	iexplore.exe	C:¥Program Files¥Internet Explorer¥iexplore.exe C:¥PROGRA~1¥INTERN~1¥IEXPLORE.EXE
Thunderbird 38.1.0	thunderbird.exe	C:¥Program Files¥Mozilla Thunderbird¥thunderbird.exe C:¥PROGRA~1¥MOZILL~1¥THUDE~1.EXE
Word	WINWORD.EXE	C:¥Program Files¥Microsoft Office¥OFFICE11¥WINWORD.EXE C:¥PROGRA~1¥MICROS~2¥OFFICE11¥WINWORD.EXE
Excel	EXCEL.EXE	C:¥Program Files¥Microsoft Office¥OFFICE11¥EXCEL.EXE C:¥PROGRA~1¥MICROS~2¥OFFICE11¥EXCEL.EXE
Adobe Reader 8.0	AcroRd32.exe	C:¥Program Files¥Adobe¥Reader 8.0¥Reader¥AcroRd32.exe C:¥PROGRA~1¥Adobe¥READER~1.0¥Reader¥AcroRd32.exe
三四郎 2010	SNS14.exe	C:¥Program Files¥JustSystems¥SNS14¥SNS14.EXE C:¥PROGRA~1¥JustSystems¥SNS14¥SNS14.EXE