ホスト型IDSを用いた標的型攻撃対策

中里　純二†　津田　侑†　高木　彌一郎†　衛藤　将史†　井上　大介†
中尾　康二†

†情報通信研究機構
184-8795 東京都小金井市貫井北町 4-2-1
{nakazato, tsuda, yaichiro, eto, dai, ko-nakao}@nict.go.jp

あらまし 標的型攻撃に利用されるツールは，アンチウィルスソフトウェアでは検知されない事が多く，攻撃に気付くことが遅れることがあり，情報漏洩などの重大なインシデントを招く恐れがある．そのため，入口対策では十分に対策は行えず，感染を前提にした対策が必要となる．これらの攻撃ではRemote Administration Tool（RAT）など，一般ユーザが通常利用するアプリケーションでは無いものが使われるため，攻撃を受けたタイミングから新規に動作が現れることが考えられる．そこで，本研究では個々のホスト内部で動作するプロセス情報（プロセスリスト）を定期的に取得し，ユーザが利用する可能性が低いプロセスを抽出する．プロセスの親子関係や実行パス（実行されたアプリケーションの保存場所）を用いた比較を行うことで，たとえ正常プロセスに成り済ましたプロセスでも親プロセスの違いや実行パスの違いから新規プロセスとして抽出する事が可能となる．その後，そのプロセスのAPI履歴などの詳細動作を取得することで悪意のあるプロセスを発見し，プロセスの動作を止めるなど迅速に対策することが可能となる．

A Countermeasure for Targeted Attacks using Host Based IDS

Junji Nakazato†　Yu Tsuda†　Yaichiro Takagi†　Masashi Eto†
Daisuke Inoue†　Koji Nakao†

†National Institute of Information and Communications Technology.
4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo 184-8795, JAPAN
{nakazato, tsuda, yaichiro, eto, dai, ko-nakao}@nict.go.jp

Abstract The APT attacks such as targeted attacks are difficult to detect by anti-virus softwares. Consequently, those attacks often lead to serious incidents such as information leakage. Therefore, it is quite reasonable for us to postulate infections by APT based attacks because of lack of detection capabilities at the border entrance. The APT attacks often use specific tools such as Remote Administration Tool (RAT) which are never used by normal users. Therefore such APT attacks by means of specific tools can be recognized by the new types of attacking behaviors (process information). In this paper, we gather the process information (behaviors) from each host to confirm if there are any new or unusual processes. The proposed system can successfully extract a suspicious process, even if the process is pretended to be a regular process using the process information that is constituted from execution path, parent-child relationship and so on. After the extraction, specific malicious process can be further detected by using API history and etc., so as to proceed to possibly kill the process in a prompt manner.

1　はじめに

標的型攻撃は，入念な計画を立てた後に多くの技術を用いて戦略的に行われる．事前に標的とする相手や組織の情報を十分に収集し，SNSやメールなどを用いて相手への侵入を試みる．このとき利用されるツールは，アンチウィルスソフトウェアによる対策を回避するなど，攻撃の事実を見つけることが困難になっている．そのため，攻撃されたこと自体に気付くことが遅れ，長時間に渡り侵入を許してしまうことになる．実際に，宇宙航空研究開発機構（JAXA）では平成 23 年 3 月 17 日に標的型攻撃（メールによる侵入）によってマルウェアに感染し，平成 24 年 11 月 21 日にアンチウィルスソフトウェアによって発見されるまで，1 年以上の間感染に気付かず，多くの情報が漏洩した可能性が高い事件が発生している [1]．この事例のように，マルウェアの侵入を防ぐ入口対策のみでは標的型攻撃に対する万全な備えになるとは言えず，マルウェアに感染されることを前提とした対策も重要となる．

さらに，標的型攻撃では，RAT（Remote Administration Tool）や OS の管理者用コマンドを始めとしたツールが複合的に利用されることが知られている．すなわち，一般ユーザが通常利用するアプリケーション以外のプロセスが攻撃の過程で生成されることになる．このような通常生成されないプロセスをいち早く検出することはマルウェア感染を前提とした対策では重要となる．

そこで，本研究では個々のホスト内部で動作しているプロセス情報（プロセスリスト）を定期的に取得し，ユーザが普段利用しないプロセスを抽出する．プロセスリストの作成時にプロセスの親子関係（プロセスパス）や実行されたアプリケーションの保存場所（実行パス）を同時に記録する．プロセスパスなどを含めることで，たとえば，正常プロセスに成り済ました不正なプロセスが実行された場合でも，実行される順序の違いからプロセスパスが異なっていたり，実行パスの違いを認識でき，通常とは異なる不審なプロセスとして抽出する事が可能となる．その後，その不審なプロセスの API 履歴などの詳細動作を取得することで悪意のあるプロセスを発見し，危険な動作を確認した場合に，即座にプロセスの動作を止めるなど迅速な対策を行うことが可能である．

本論文では，実際に 4 人のユーザと擬似的に不正プロセスを実行するテストユーザ 1 人に対して，不審プロセスの生成について調査を行った結果を報告する．実行された全 518 プロセス中 218 プロセスが不審プロセスとして抽出され，擬似的に実行した不正プロセスを正しく抽出できた．

本論文の構成は以下の通りである．2 節では，関連する標的型攻撃に対する対策技術の紹介を行う． 3 節では，実際にユーザホストからプロセスの情報を取得するシステムの説明を行い，取得した情報から不審プロセスを抽出する方法を 4 節に示す．また，実際に 5 人のユーザから取得した結果を示し，最後に 5 節でまとめる．

2　関連研究

標的型攻撃の対策を大別すると，攻撃の侵入を防ぐ入口対策，万が一侵入を許してしまった場合でもそれ以上の被害拡大を防ぐ内部対策，重要な情報などが外部に漏洩することを防ぐ出口対策の 3 つがある．入口対策では，アンチウィルスソフトウェアなどにより悪意のあるソフトウェアの侵入を防ぐ．しかし，特に標的型攻撃では，アンチウィルスソフトウェアによる対策を回避するなど，侵入を完全に防ぐことは難しい．そこで，内部対策や出口対策が非常に重要となっている．

2000 年始めより，有限責任中間法人 JPCERT コーディネーションセンター（JPCERT/CC）[2] や独立行政法人情報処理推進機構（IPA）[3] などによる標的型攻撃調査が行われている [4, 5, 6]．情報窃取を目的とした攻撃では，攻撃者のリスクを低減する為にステルス性を高め確度の高い攻撃が行われることが示されている．ステルス性を高めることにより，攻撃者にとって攻撃が見つかるリスクを抑え，さらに長期的な攻撃（情報窃取）が可能になるとされている．

文献 [7] では，メールなどに添付されて侵入

を試みる攻撃に対する対策を提案している．遠隔操作されたコンピュータから標的型攻撃メールを送信する場合に，事前に抽出した本人の行動特性と攻撃者によって操作されたときの特性を比較し，送信されたメールが攻撃か否かの判別を行う方式の提案を行っている．

内部対策では，標的型攻撃を受けて感染した犠牲ホストが行うネットワーク活動に着目した対策が多く研究されている [8, 9, 10]．文献 [8] では，送信元 IP アドレスや送信先 IP アドレス，宛先ポート番号に従って抽出した時系列の特徴からデータの傾向の変化を捉え，不審な通信を抽出する．従来方式である ChangeFinder[11] に比べ早い段階で不審な通信を発見することが可能となっている．文献 [9, 10] では，攻撃基盤を拡大する過程で SMB を悪用して次の標的を探すノード探査や，標的のノードのリモート制御などの攻撃者が使わざるを得ない／内部的に共通して使われている攻撃手法（チョークポイント）に着目している．チョークポイントを利用することで，システムの振る舞いに矛盾・異常がないかを判定し，正規プログラムに成りすました攻撃や正規通信に紛れる攻撃，亜種・未知・難読化などのアンチウィルスソフトウェアを回避するマルウェアによる攻撃を検出することに成功している．

3　提案システム

3.1　システム概要

提案システムの全体概要を**図 1** に記す．本システムでは，監視対象セグメントにある各ホストにインストールしたエージェントから各種情報を収集し，管理サーバで保存・分析を行う．その結果の可視化等を行うことで，脅威の発見を迅速に行えるようオペレータに対して支援する．また，その他の異常検知エンジンの情報からエージェントに対してフィードバックを行い，重大なインシデントが発生する前に迅速に対策を行うことが可能となる．

本論文では，特にエージェントが収集するプロセス情報（プロセスリスト）に注目した分析を行う．

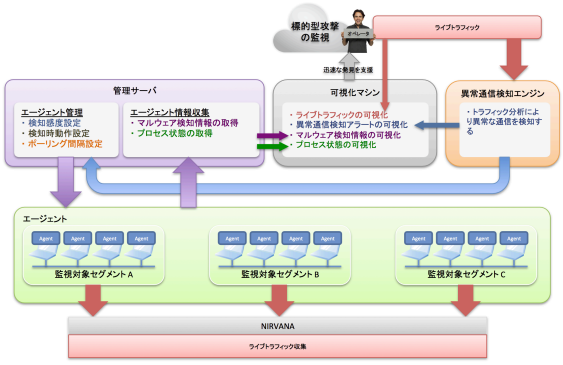

図 1: システム概要

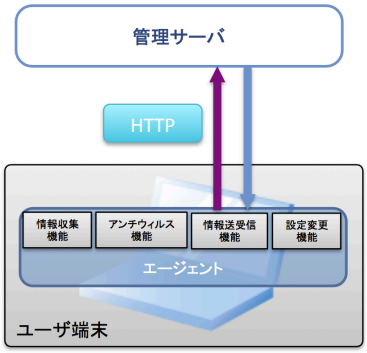

図 2: エージェント概要

3.2　機能概要

ここでは，各ホストにインストールするエージェントツールの機能を説明する．**図 2** にエージェントの概要を示す．エージェントは，1) 情報収集機能，2) アンチウィルス機能，3) 情報送受信機能，4) 設定変更機能の 4 つの機能からなっている．

1) **情報収集機能：** 監視対象となっているプロセスの動作を定期的に記録する．具体的には，CPU 使用率，メモリ使用率，プロセス実行パス（プロセスが実行された場所），プロセス ID，親プロセス ID，ネットワーク状況の情報を取得する．親プロセス ID とは，当該プロセスを実行したプロセスのプロセス ID を表し，この情報によりプロセスツリー（プロセスの親子関係）を再現することが可能となる．また，指定されたプロセスが利用した API 履歴を取得し，ホスト内に保存する．

2) アンチウィルス機能： 一般的なアンチウィルス機能により，不正プログラムの検出・停止などを行う．また，管理サーバからの指示により，不正プログラムの検出のみを行い，プロセスの停止を行わない,「泳がせモード」を有している．特に，標的型攻撃などでは，実際の攻撃の詳細な手口が未だに明らかとなっていないなど，一定の動作を観測する機構が重要となる．継続的に攻撃者の挙動を分析することで迅速な対策に結びつけることが可能になる．一方で，情報の持ち出しなど本当に危険な動作などを検知した場合は，特定のプロセスを指定して停止を行う，プロセス停止機能も有している．

3) 情報送受信機能： 情報送受信機能では管理サーバに対して，マルウェア感染時などに情報をリアルタイムに送信する,「a. マルウェア検知報告機能」，定期的にホストの状態を送信する「b. プロセス状態報告機能」，管理サーバからの応答や設定情報を受信する「c. 情報受信機能」を有する．以下にそれぞれの機能の説明を示す．

a. マルウェア検知報告機能
マルウェア検知報告機能では，マルウェアの感染の可能性がある不正な動作を検知したタイミングで，管理サーバに対して当該プロセスに関する情報を報告する．情報には，ホストの識別子，検知理由，当該プロセスの状態（停止 or 実行中など），検知時に動作していたプロセスリスト，検知プロセスが利用していた通信情報が含まれる．また，感染検知時の OS イベントログやメモリダンプ結果をホスト端末内に保存する．

b. プロセス状態報告機能
アンチウィルス機能で発見することが難しいマルウェアの動作等を迅速に発見するため，定期的に管理サーバに対し，ホスト内で動作しているプロセスの状況を通知する．このとき，ホスト ID,「1) 情報収集機能」で収集した情報を送信する．また，API 履歴の取得を指定されたプロセスに関しては，API 履歴の概要（統計情報）を送信する．

c. 情報受信機能
管理サーバからの応答を受信し，必要に応じて「4) 設定変更機能」により設定を動的に変更を可能にする．管理サーバからの応答には，情報を正しく受信したことを示す受信確認（ACK），「2) アンチウィルス機能」の感度変更メッセージ,「b. プロセス状態報告機能」の送信間隔（ポーリング間隔）の変更メッセージ，不正プログラムの検出時にプロセスの停止を行うかを指定する検知ロジック変更メッセージ，API 履歴の取得を指示するメッセージの 5 つのメッセージがある．

4) 設定変更機能：「3) 情報送受信機能」によって受信したメッセージに従い，各機能のパラメータをエージェントの再起動などを必要とせずに動的に変更する．

3.3 ポーリング情報

本システムでは，エージェントにより取得した情報を定期的に管理サーバへ HTTP を利用してエージェントから定期的に情報を送信する．ポーリング方式にすることで，ファイアウォールなどのネットワーク環境による影響を最小限にし，エージェントと管理サーバ間の接続性を確保する．

各エージェントは，ポーリングにより送信する情報を**図 3** に示す XML フォーマットにより管理サーバへと送信する．**図 3** では，2014 年 8 月 10 日の 18:00 に行われたポーリングの結果を示している．`<process_information_list>`タグ以下に，当該ホストで動作しているプロセスリストが記録されている．この例では，プロセス ID が 568（`parent_id="568"`）のプロセスから呼ばれた，プロセス ID が 2,016（`id="2016"`）の `CcmExec` プロセス（`name="CcmExec"`）が動作していることが分かる．`CcmExec` プロセスは，`C:\Windows\System32\CCM\CcmExec.exe` から

```
<?xml version="1.0" encoding="utf-8"?>
<nirvana_request message_type="1" version="2" request_datetime="2014-08-10 18:00:00">
  <host_information id="a799ebba9388...cb825ae9d">
    <!-- ネットワークインターフェース情報 -->
    <network_interface macaddr="**:**:**:**:**:**">
      <ipaddress addr="***.***.***.***"/>
    </network_interface>
    <!-- 利用ユーザ名 -->
    <logon_user user="******"/>
    <!-- システム情報 -->
    <os_information os="Windows 7" service_pack="Service Pack 1" architecture="x86"/>
  </host_information>
  <!--既にマルウェアの検出が行われているか-->
  <detected_state detected="true"/>
  <!-- プロセス情報 -->
  <process_information_list>
    <process_information id="2016" name="CcmExec" cpu="0.0" mem="34148352"\
                         status="Execute" parent_id="568" path="C:\Windows\System32\CCM\CcmExec.exe">
      <tcp_state ip_src="***.***.***.***" port_src="49296" ip_dst="***.***.***.***"\
                                                          port_dst="80" state="ESTABLISHED"/>
      <udp_state ip_src="127.0.0.1" port_src="58798"/>
    </process_information>
          :
  </process_information_list>
          :
</nirvana_request>
```

図 3: プロセスの状態報告 XML 例

実行され，ネットワーク接続を行い，49,296 番ポートから 80 番ポートに接続している．さらに，58,798 番ポートで UDP の接続待ちを行っていることが確認できる．

4 提案方式

本節では，エージェントツールにより取得するホスト情報から，不審プロセスの抽出を行い，不正プロセスを特定する方式の説明を行う．標的型攻撃では，日常的にユーザが利用するプロセス以外のプロセス（例えば RAT など）が動作することが考えられる．そこで，ユーザごとに利用するプロセスのプロファイリングを行い，新規実行されるプロセスを抽出する．他のユーザでもそのプロセスが実行されたことが無く，真の新規プロセスであった場合，それを不審プロセスとする．さらに，その不審なプロセスの API 履歴などの詳細情報を取得することで，RAT などの不正プロセスの特定を行うことが可能である．

4.1 不正プロセス抽出方法

Step 1: プロファイリング エージェントが各ホストから取得するホスト情報，特にプロセス情報を抽出し，ユーザごとにプロセスのプロファイリングを行う．プロセスのプロファイリングには，プロセス ID を元に再構成したルートプロセスからのプロセスパスを保存する．Internet Explore を用いて PDF を閲覧した場合プラグインが動作し，root-explorer-iexplore-Acrobat というプロセスパスが取得できる．このとき，プロセスの親子関係のみならず実行パスも同時に記録することで正規プロセスに成り済ました場合でも，実行パスやプロセスパスが異なると，別プロセスとして検出が可能になる．一定期間，ユーザが利用するプロセスをプロファリングすることにより，ユーザが日常的に使うプロセスか否かを確認する．

Setp 2: 不審プロセスの抽出 ユーザが今まで利用したことの無いプロセスパスを持つプロセスを実行した場合，そのプロセスが他のユーザによって利用されているか（同一のプロセスパスを持つプロセスが実行されているか）を比較する．もし，他のユーザ

表 1: 実験環境

CPU	Core2 Duo 3.0 GHz
メモリ	4 GB
HDD	500 GB
OS	Windows 7 (32 ビット)

含め始めて実行されたプロセスであった場合，不審なプロセスとして API 履歴の取得など詳細情報の取得を行う．

Step 3: 詳細調査 API 履歴などの結果から，当該プロセスが不審な動作を行っていないかを調査する．特に，システム情報を取得する API や，外部接続を行う API の利用などを多用している場合，索敵や収奪など標的型攻撃に代表される危険な動きを行っている可能性も高く，注意する必要がある．

Step 4: 不正プロセス検出 明らかに悪意のある動作が確認された場合，不正プロセスとして検出する．

Step 5: 停止 危険なコマンドや API を利用したことが確認された場合，強制的にプロセスを停止し被害を最小限に食い止める．

4.2 不審プロセスの抽出実験

ここでは，不審なプロセスの抽出実験を行う．実際に NICT の情報システム室が管理を行っている PC に対してエージェントツールを導入した．情報システム室で管理を行っている PC では，一般ユーザ権限で利用可能なソフトウェア以外のソフトウェアのユーザによるインストールが許可されていないため，個々にインストールされているアプリケーションは大きく違わない特徴がある．**表 1** に実験に用いたハードウェア環境を示す．

2014 年 8 月 1 日から 2014 年 8 月 15 日までの約 2 週間のプロセス情報を取得し，新規プロセスの出現頻度を調査した．同期間にアクティブとなったユーザは実ユーザ 4 人と疑似的な不正ソフトウェアを動作させたテストユーザ 1 人の計 5 ユーザであった．擬似的な不正ソフトウェアは，ローカルネットワークにスキャンを行う新しいプロセスを使用した．

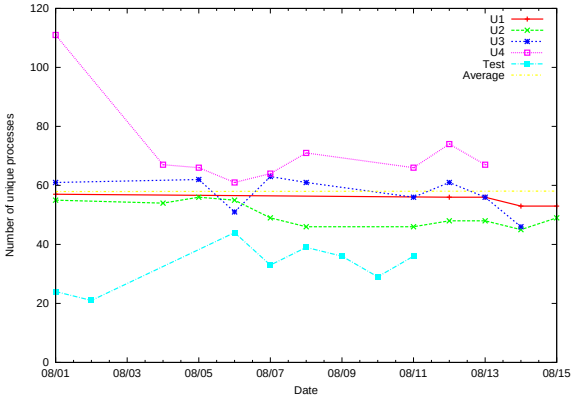

図 4: ユーザごとのプロセス数推移

4.2.1 プロセス数推移

図 4 に各ユーザの日ごとに実行したユニークなプロセス数の推移を示す．ここで，U1, . . ., U4 はアクティブな 4 人のユーザを示し，Test は，テストユーザを示す．テストユーザは最小限のプロセスしか動作させていないため，相対的な実行プロセス数が少なくなっている．そのため，平均プロセス数（Average）にはテストユーザのプロセス数は含んでいない．**図 4** より，各ユーザの最大プロセス数は U4 の 111 プロセスであり，最小プロセス数は U2 の 45 プロセスであった（テストユーザのプロセス数を除く）．各ユーザとも，平均すると約 60 プロセスが実行されていることが分かった．

4.2.2 新規プロセス数推移

図 5 に各ユーザが日ごとに 1 度だけ実行したプロセス数の推移を示す．ここで，U1, . . ., U4 は 4.2.1 節と同様にアクティブな 4 人のユーザを示し，Test は，テストユーザを示す．テストユーザは最小限のプロセスしか動作させていないため，新規プロセス数も相対的に少なくなっている．そのため，平均プロセス数（Average）にはテストユーザのプロセス数は含んでいない．**図 5** より，各ユーザの最大プロセス数は U4 の 61 プロセスであり，最小プロセス数は同じく U4 の 19 プロセスであった．各ユーザとも，平均すると約 40 プロセスが 1 日に 1 度のみ実行されていることが分かった．

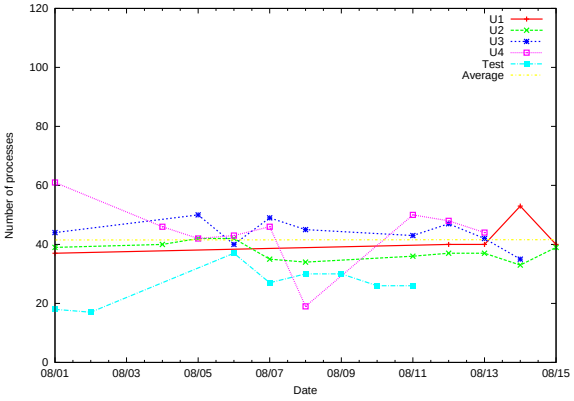

図 5: ユーザごとの新規プロセス数推移

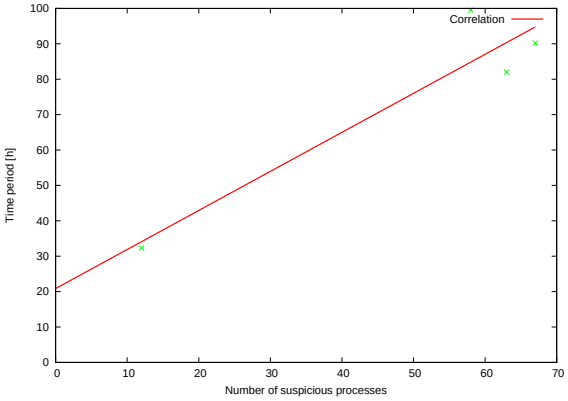

図 6: 起動時間と不審プロセス数の関係

表 2: 新規プロセス比較結果

ユーザ	U1	U2	U3	U4	Test
全プロセス数	106	169	218	226	102
新規プロセス数	29	80	106	104	38
不審プロセス数	12	58	63	67	18

図 4，**図 5** より，ユーザが利用するプロセスは，40 プロセス程度が 1 日に 1 回のみ実行されていることから，20 プロセス程度が 1 日に複数回実行されていることが分かる．従って，多くのプロセスは，1 日に 1 度のみ実行されていることが分かる．

4.2.3 不審プロセスの抽出

図 5 は，ユーザごとの新規プロセス数（調査期間中の同日中にそのユーザでは 1 度のみ実行されたプロセス数）の推移である．しかし，プロセスによってはあるユーザでは新規プロセスであるが，既に他のユーザが実行していることも考えられる．特に，ソフトウェアのアップデートを行うプロセスなどは，頻繁には実行されないため，ユーザごとには新規プロセスとなる可能性が高いが，他のユーザでも実行されることが十分に考えられる．そこで，ユーザ間でのプロセスパス比較を行い，全ユーザでの新規プロセスを抽出する．全体で 1 度しか実行されないプロセスは，不審なプロセスの可能性が高く，詳細な情報を取得する必要がある．**表 2** に，各ユーザの新規プロセス数と不審プロセス数を示す．

表 2 より，各ユーザの実行プロセスのうち約 40% が新規プロセスとなることがわかった．さらにその約 30% が不審プロセス（全体で 1 度のみ実行されたプロセス）となることが分かった．また，ユニークなプロセスパス数は 518 種類あり，一度しか実行されていないプロセスは 218 プロセスであった．**図 6** にユーザの利用時間（ホストの起動時間）と不審プロセス数の関係を示す．図より，ホストの起動時間が長いほど不審プロセス数が多くなる傾向がある．

Test ユーザでは，実際に擬似的な不正プロセスの実行を 2014 年 8 月 6 日に行った．**図 4**，**図 5** より，Test ユーザのユニークプロセス数，新規プロセス数は 8 月 6 日には共に最大値である 44 プロセスと 37 プロセスを記録している．実験期間中（8 月 1 日から 15 日）に Test ユーザが 1 度のみ実行したプロセス数は 38 プロセスであった．このうち，他のユーザが実行した経験のあるプロセスは 20 プロセスあった．従って，Test ユーザのみ 1 回だけ実行したプロセスは 18 プロセスあった．このうち，1 つは擬似的に実行した不正プログラムを含んでいた．

4.2.4 考察

ユーザ単独のみならず，ユーザ間のプロセス比較を行うことで，不審なプロセスの抽出を行った．特に Test ユーザで実行した擬似的な不正プログラムを正しく抽出することが可能であった．一方で，擬似的な不正プロセス以外にも 17 プロセスが不審プロセスとして抽出された．これ

表 3: 誤検知プロセスの詳細

種類	内容	プロセス数
アップデート関連	Adobe Flash, RealPlayer など	4
システム関連	ドライバなど	8
アプリケーション特有	プラグインなど	5

らのプロセスは，不審プロセスの可能性は低い，アップデートプロセスや Windows のシステムプロセスであることが分かった．**表 3** に不審なプロセスと誤検知されたプロセスの詳細を示す．これらのプロセスは，アップデートプロセスや，ドライバ関連のプロセスが多く含まれていた．これは，プロファイル作成期間が短く，さらにユーザの夏期休暇期間と重なっていたため，十分なデータが取得できなかった可能性が高い．今後，さらにプロファイル期間を延ばし，十分なデータを利用した分析を行う必要がある．また，API 履歴の取得などを行うことで，不正なプロセスでは無いことが判明する可能性も高い．

5 まとめ

本論文では，エージェントツールにより収集したホストの状態，特にプロセスの状態を用いて不審なプロセスの抽出を行った．実験期間中に実ユーザ 4 人と擬似的な悪性プロセスを実行するテストユーザ 1 人を含めた 5 ユーザが 518 プロセスを実行した．そして，218 プロセスが全てのユーザ間で 1 度しか実行されない不審なプロセスであると判定された．テストユーザで実行した不正プロセスは，不審なプロセスとして正しく抽出された一方で，正規なプロセスの可能性が高い 17 プロセスも不審なプロセスと判定された．これらのプロセスは，アップデート用プロセスや，プラグインなどの特定のタイミングでのみ実行されるプロセスであった．そのため，プロファイリングを行った期間が短く十分なデータが得られなかったために誤判定した可能性が高い．

不審なプロセスが発見された場合，そのプロセスの詳細情報（API 履歴など）を取得するなど監視下に置くことで，不正な動作（不正な API 利用や，不審なネットワーク接続など）をいち早く捉え，迅速な対応を行う必要がある．

今後，より多くのデータを用いた不審プロセスの抽出を行い，API 履歴などの詳細情報を取得した，精度高い不正プロセスの特定を行い，より迅速な対策を行う．

参考文献

[1] 宇宙航空研究開発機構プレスリリース, “JAXA におけるコンピュータウィルス感染に関する調査結果について”, http://www.jaxa.jp/press/2013/02/20130219_security_j.html （2014 年 8 月現在）

[2] 有限責任中間法人 JPCERT コーディネーションセンター（JPCERT/CC）, http://www.jpcert.or.jp/（2014 年 8 月現在）

[3] 独立行政法人 情報処理推進機構（IPA）, http://www.ipa.go.jp/ （2014 年 8 月現在）

[4] JPCERT/CC, “標的型攻撃について” （PDF）, https://www.jpcert.or.jp/research/2007/targeted_attack.pdf （2014 年 8 月現在）

[5] JPCERT/CC, “標的型攻撃対策手法に関する調査報告書”（PDF）, http://www.jpcert.or.jp/research/2008/inoculation_200808.pdf （2014 年 8 月現在）

[6] IPA, “標的型サイバー攻撃の事例分析と対策レポート”, https://www.ipa.go.jp/security/fy23/reports/measures/index.html （2014 年 8 月現在）

[7] 片山 佳則, 寺田 剛陽, 津田 宏, “利用者の行動特性を用いたサイバー攻撃における成りすまし対策技術”, 人工知能学会全国大会, 4G1-4, 2014.

[8] 北澤 繁樹, 祢宜 知考, 河内 清人, 榊原 裕之, 藤井 誠司, “標的型攻撃検知システムの評価”, マルウェア対策研究人材育成ワークショップ（MWS 2009）, A6-3, 2009.

[9] 海野 由紀, 森永 正信, 山田 正弘, 鳥居 悟, “標的型サイバー攻撃におけるシステム内部の諜報活動検知の提案”, コンピュータセキュリティシンポジウム（CSS 2012）, pp. 360 – 367, 2012.

[10] Satoru Torii, Masanobu Morinaga, Takashi Yoshioka, Takeaki Terada, and Yuki Unno, “Multi-layered Defense against Advanced Persistent Threats (APT),” FUJITSU Sci. Tech. J., Vol. 50, No. 1, pp. 52 – 59, 2014.

[11] Jun-ichi Takeuchi and Kenji Yamanishi, “A Unifying Framework for Detecting Outliers and Change Points from Time Series,” IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Issue 4, pp. 482 – 492, 206.