「新しいタイプの攻撃」への対策支援と，そのタスクスケジューリングの手法を用いた並列分散処理化手法の提案

松浪拓海[†1] 小出洋[†2,†3]
金岡晃[†4] 加藤雅彦[†5]

企業や政府等の組織の内部情報網は，ここ数年で急速に電子化を進め，秘密情報を含むあらゆる情報は電子的に管理，保管され，組織内での情報伝達もメールや独自ポータルサイトによって電子的に行われることが多くなっている．このことから，情報の窃取を企む組織の窃盗手法も変化し，最近は電子的に情報を盗み出そうとする事例と，それによる秘密情報の盗難被害が増加している．とりわけ，APT（Advanced Persistent Threats）と呼ばれる，特定の組織や個人を標的とし，確実に情報を盗み出すための執拗で高度な，情報システムに対するサイバー攻撃が問題となっている．APT は独立行政法人情報処理推進機構（IPA）では「新しいタイプの攻撃」という表記にて呼称され，2010 年 12 月に発足した「脅威と対策研究会」によっても問題の分析とその解決策の提案が行われている．複雑で事例のそれぞれが特殊なこの種のサイバー攻撃の挙動は，セキュリティ分野のエキスパートによって事例毎に分析されているものの，現在の情報システムは複雑で大規模なために，人間が網羅的でかつ見落としの無い分析を効率的に行うのは難しい．そのため，情報システムにおける「新しいタイプの攻撃」による脅威の挙動を計算機で網羅的にトレースし，エキスパートの専門知識と経験に基づく分析を補う必要があると考えられる．本予稿では，この「新しいタイプの攻撃」による脅威の挙動のトレースを，並列分散処理のためのタスクスケジューリングの手法により並列分散処理化し，効率的に実行するためにどうすべきかについて提案し，議論したい．

A Proposal of Measure Support for APT and it Parallel Distributed Processing by Task Scheduling Method

Takumi Matsunami,[†1] Hiroshi Koide,[†2,†3] Akira Kanaoka[†4] and Masahiko Katoh[†5]

In these years, organizations, enterprises and governments, store and manage almost all information including confidential information to their cyberspaces rapidly. They also exchange them in their organizations by using E-mail and their portal site. Therefore, strategies of criminal organizations which try to steal confidential information have changed. Lately, they bring out it electronically and cases of information abstraction are increasing. Especially, we have to pay attention to cyber attacking threats which target to specific organizations and persons. This type of cyber attacking threats are called APT (Advanced Persistent Threats). In this study, we present a threat trace method and apply parallel and distributed techniques including task scheduling to it. Many experts in cyber security field have analyzed each behavior of APT which is complicated and customized. However it is difficult that we analyze behaviors of APT comprehensively, because current information systems are too complicated and too large. Therefore, we propose the threat trace method which simulates behaviors of threats exhaustively to complement the analysis by the experts based on the expertise and experience. In this proceedings, we suggest threat trace method and applying it parallel distributed processing by task scheduling method for parallel distributed processing and discuss about it.

†1 九州工業大学大学院情報工学府
Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology
†2 独立行政法人情報処理推進機構
Information-technology Promotion Agency, Japan
†3 九州工業大学大学院情報工学研究院
Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology
†4 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†5 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc.

1. はじめに

本研究の目的は，一般に APT（Advanced Persistent Threats）と呼ばれている，組織の情報システムに対する脅威によるサイバー攻撃の成功を未然に防ぎ，情報システムの電子的破壊や内部情報の漏洩等を防ぐための手段を明らかにすることである．

近年，企業や政府等の組織では内部情報網の電子化が急速に進んでおり，組織内部で扱う情報を電子的に流通させ，また保管することによって，利便性が大きく高められている．それに伴って，およそ組織にとって秘匿したい様々な秘密情報も，同様に電子的に保管され，管理されるようになっている．このことから，組織内にて保管されている情報の窃取を図る組織が情報の窃取に用いる手法も変化しており，昨今は APT によるものが増えている[1]．特に，企業活動において極めて重要な製品設計の情報や顧客に関する情報，政府の中央省庁における外交上の秘密や防衛に関わる機密情報等は，情報窃取の対象として狙われやすい．このような情報は，漏洩すると方々に損害を与えることとなるが，電子的に保管・管理されているが故に，一度漏洩すると取り戻すことはできない．また，情報システム自体も組織の活動においては極めて重要なものになっており，システムの停止は業務の停止に直結するため，情報システムの破壊は組織の活動を一時的に停止させ，損害や不都合を発生させるなどして信用を低下させる．このような状況は，効果的な対策を講じることによって防がなければならない．

本予稿では，組織の情報システムに対する APT による攻撃の成功を未然に防ぎ，マルウェアの脅威による情報システムの電子的破壊や内部情報の漏洩等を防ぐための「脅威トレーサ」を提案する．また，この脅威トレーサを実装するにあたっては，並列分散処理のためのタスクスケジューリングの手法を用いた，複数の計算機による効率的実行が必要であると考え，タスクスケジューリングの手法を用いて情報システムの模擬を並列分散処理化するためにはどうすべきかということについて議論する．

2. 新しいタイプの攻撃

APT（Advanced Persistent Threats）とは，既知あるいは未知の脆弱性を悪用し，既存の攻撃手法を複数組み合わせ，ソーシャルエンジニアリングの手法を利用するなどして，特定の組織や個人を執拗に狙う脅威を指す言葉である．従来の愉快犯的な情報システムへの攻撃と異なり，情報の窃取やシステムの破壊など，明らかな目的を持ち，その主体も個人あるいは小規模な組織から，複数の攻撃者が連携しあるいは国家レベルで協力をしている大規模な組織によるものへと変化してきている．独立行政法人情報処理推進機構（IPA）では，この APT を「新しいタイプの攻撃」と表記しており，2010 年 12 月に「脅威と対策研究会」を発足して，APT の分析と対策についての研究を行っている[3]．

2.1 攻撃の特徴

「新しいタイプの攻撃」によく見られる特徴として，

- ある特定の組織や個人に標的を絞っている，
- 執拗でかつ継続的に，密かに攻撃を行い続ける，
- 攻撃対象の情報システムの特徴と防御策に対応して攻撃手法を変化させる，
- ソフトウェアのセキュリティ上の脆弱性のうち，一般には未発見のものを突いた攻撃（ゼロデイ攻撃）によることが多い，
- 攻撃の痕跡を残さず，特に攻撃に使われたマルウェアを取り押さえることができない，

ということが挙げられる．

攻撃は，標的型メールの添付ファイルや悪質な Web ページからのダウンロードによる，情報システム内部へのマルウェアの侵入から始まる．攻撃対象の情報システムに侵入したマルウェアは，情報システムのネットワークにバックドアを生成し，攻撃者と連絡を密に取り合いながら，攻撃の準備を進める．このとき，攻撃者はマルウェアをアップデートすることが可能であり，マルウェアが収集した攻撃対象の情報システムの構成や，攻撃に対する防御策に関する情報を受信して，その情報に基づいて適宜マルウェアをアップデートすることにより，情報システムのより深い部分への侵攻を試みることもある．このように，「新しいタイプの攻撃」は，特定の組織や個人の情報システムを標的とした攻撃で，求める情報を探し当て，確実に盗み出そうとする[2]．

2.2 入口対策と出口対策

従来行われてきたセキュリティ対策として，ウイルス対策ソフトウェアの導入と更新，OS や業務ソフトウェア等へのセキュリティパッチの適用，基幹ネットワークへのファイアウォールと侵入検知システムの導入等がある．これらの対策は，いずれもマルウェアや攻撃者による侵入行為を，組織の内部に到達させないための対策であるというところに共通点がある．これを「入口対策」という．しかし，ウイルス対策ソフトウェアに検出できない未知のウイルスや，OS や業務ソフトウェア等に内在する，広く知られておらず対策

の施されていない未知の脆弱性は存在し，上記のような対策をとっていたとしても，この脆弱性を突いた攻撃（ゼロデイ攻撃）によりマルウェアや攻撃者に侵入されることがある．また，組織内で行われる情報モラル教育によって，「怪しいメールは開かない」，「怪しい Web ページにはアクセスしない」というような教育が行われることはあるが，昨今の業務ではメールや Web を頻繁に活用することはもはや当たり前であり，どのように気をつけていたとしても，いつかは開いてしまうものと考えられる．

「新しいタイプの攻撃」においては，最初の攻撃は組織内部の端末利用者を不正プログラムをダウンロードさせるような Web ページへ導入したり，標的型メール攻撃のターゲットとすることから始まる．このときに導入されるマルウェアは非常に小さいものであるが，最初の一つを情報システム内部に取り込ませてしまうということに意味がある．このことから，入口対策のみによるセキュリティ対策には限界があるということが分かる．

そこで，マルウェアが情報システム内部に侵入してしまうということを認めた上で，攻撃者がそれ以上の攻撃行動をすることができないように，マルウェアによる活動を抑止したり，秘密情報を持ち出せないようにしたりするための対策が必要である．これを「出口対策」という．出口対策の必要性については，内閣官房情報セキュリティセンター（NISC）がリスク要件リファレンスモデル[4] として研究をまとめているほか，IPA も「新しいタイプの攻撃」への重要な対策として主張している．

3. 提 案 手 法

本研究では，組織の情報システムと，これに侵入するマルウェアとをモデル化し，計算機によってマルウェアの挙動を追跡する「脅威トレーサ」を提案する．この脅威トレーサは，マルウェアによる脅威を計算機上のモデルによりシミュレートする．このシミュレーションを，「脅威トレース」という．本研究の提案する手法では，脅威トレーサによる脅威トレースの結果から得られた情報を元にして，本研究の目的である，「新しいタイプの攻撃」による情報システムの電子的破壊や内部情報の漏洩等への対策を支援し，これら防ぐための手段を明らかにする．

3.1 既 存 研 究

マルウェアのモデル化の部分に関する既存研究としては，米国の The Mitre Corporation[5]（以下，MITRE と表記）による MAEC[6]（Malware Attribute Enumeration and Characterization）が挙げられる．MAEC は，多くの亜種を含む，世の中に出回る多くのマルウェアを分類し，整理して，それぞれの特徴付けをするための標準的な方法を提供している．また，情報システムのモデル化の部分に関しては，金岡ら[8] が NSQ（Networked-system Security Quantification）モデルを提案している．

MAEC によるマルウェアのモデルは，あらゆるマルウェアの亜種を分類できるように，ほかのマルウェアとのわずかな相違も詳細に記述し，書き分けることができる．しかし，MITRE の目的は，マルウェアの情報を詳細に記録するための標準的なフォーマットを MAEC によって提供し，マルウェアを決まったタイプに明確に分類して系譜を追跡し，正確にグループ化，記述することにより，ある種のマルウェア・データベースを構築することである．そのため，本研究で提案する脅威トレーサでそのまま用いるには冗長な部分があり，また，情報システムのネットワーク上にある複数の計算機が互いに影響し合うというような挙動に関する記述をする部分に関しては不足している．

3.2 ドメイン固有言語

このため，小出ら[7] は，NSQ モデルを基本としたデータ構造によって情報システム全体のスナップショットを表現し，その上でマルウェアの挙動をモデル化するドメイン固有言語（Domain Specific Language; DSL）を提案している．この DSL では，情報システムとマルウェアの挙動を適当な抽象度で抽象化するための定義をしている．たとえば，ルータ 1 台を介して 2 台の PC が接続されてネットワークを構成する単純な情報システムは，DSL を用いると図 1 のように表される．

「新しいタイプの攻撃」に用いられるマルウェアは，対象を 2 段階で攻撃しようとするものと考えられている．まずは攻撃対象に侵入するために，システム内部で端末を操作する人が興味を持ち，メールや Web サイトへのリンクを開いてしまうように仕向けるものや，従来の不正アクセスの手法（SSH アタック等）によるもの，またはそれらの侵入行為が有利に働くように，ソーシャルネットワークを利用して興味と趣向を探るものもある．これらへの対策は「入口対策」にあたり，従来のウイルス対策ソフトウェアや侵入検知システム，情報モラル教育等の導入と，これらのさらなる発展が望まれる．

マルウェアがシステム内部に侵入すると，まずマルウェアは次のようにして攻撃の基盤を構築しようとすると考えられる．

```
object Sample extends MalwareSimulationLibrary {
  val pc1, pc2 = PCterminal
  val router1 = Router
  val malware1 = Conficker
  val rtable1 = RoutingTable

  pc1 has a networkCard ‘‘192.168.0.2/24’’
  pc2 has a networkCard ‘‘192.168.1.2/24’’
  pc1 opens tcpPort (22, 80)
  pc2 opens tcpPort (22, 80, 443)

  router1 has a networkCard ‘‘192.168.0.1/24’’
  router1 has a networkCard ‘‘192.168.1.1/24’’
  router1 has a rtable1

  pc1 connects to router1
  pc2 connects to router1
  pc1 is infectedWith malware1

  run
}
```

図 1 単純な情報システムの DSL による記述例.
Fig. 1 An example of simple network written by DSL.

(1) システム内部のネットワーク構成を捜査し，その情報を攻撃者の元へと送るため，C&C（Command and Control）サーバとのコネクションを持つために，バックドアを生成する.
(2) ネットワーク上の他の端末への感染拡大を計ったり，攻撃者からの C&C サーバからの指示に従い，自身をアップデートしたりする動作をする.

この後マルウェアは，構築した攻撃基盤を利用して，秘密情報の探索や認証情報の盗聴，組織の Web サイトの改ざん，ほかの組織のサーバへの攻撃（の踏み台として使う）等の動作を行うと考えられる.

ここで，例えばマルウェアが攻撃の基盤を構築しようとするときの動作を DSL を用いて表現すると，図 2 のようになる．`initialize` 節はマルウェアに感染した最初の 1 台が実行する動作，`control` 節は C&C サーバからの指示を受け，指示の内容に従ってメソッドを起動する動作を示す.

3.3 脅威トレーサ

脅威トレーサは，DSL によってモデル化された情報システムとマルウェアを解釈し，情報システム上でマルウェアがどのように動作するかをシミュレートする．マルウェアは，前節で説明したように，バックドアを作り，C&C サーバとのコネクションを確立することから始まり，ネットワークの設定を不正に操作したり，ネットワーク上の他の端末に自身のコピーを送るなどして，攻撃を進行する．このとき，攻撃を進めるための操作を行う前後と，操作が成功したか失敗したか，

```
class Sample extends SimpleMalwareLibrary {
  initialize {
    makeBackDoor()
    communicatesWith(cncServer)
  }

  control {
    case Invoke => invokeMalwareCode()
    case GetNetInfo => getsNetworkInfo()
    case HttpMsgCom => communicatesWith(cncServer)
    case Updates => updatesMyselfBy(cncServer)
    case Infection => infectsTo(pc2)
  }
}
```

図 2 攻撃基盤を構築するマルウェアの DSL による記述例.
Fig. 2 An example of simple malware that build attacking platform written by DSL.

情報システムにどのような影響を与えられたかによって，そのときの状態が変化する．これらは離散状態で決定的であり，探索によってマルウェアが情報システムをどのように操作されるかを知ることができる.

脅威トレーサの実装は，DSL の記述に向き，情報システムとマルウェアの構成を記述するのに用いる XML を強力に扱えるという特徴を活かして，Scala[9] を用いることにしている．また，トレーサは，マルウェアが情報システムに与える影響を離散的なイベントとし，離散イベントシミュレーションを行う．この離散イベントシミュレーションを行う離散イベントシミュレータの実装は，Spoon らの著書[10] にあるものをベースに拡張する.

3.4 脅威トレーサの並列分散処理化

この脅威トレースの複雑さは，情報システムのネットワーク上のノードとエッジの数と，侵入するマルウェアが有する機能の数によって決定される．また，複数のマルウェアが侵入し，それぞれが相互に作用し合う可能性もあるため，これも複雑さに影響を与える.

ネットワーク上のノードは，たとえば次のようなものが考えられる:

- PC
- ルータ
- スイッチ
- ロードバランサ
- ファイアウォール
- 侵入検知システム
- ゲートウェイ型ウイルス対策アプライアンス
- 各種サーバ等

また，ここでいうネットワークのエッジとは，ノード間をつなぐネットワークケーブル等のことを指す．物理的なネットワークのトポロジによらず，NSQ モデ

ルにより階層的にネットワークを分析すると，物理的なネットワークが疎に構成されていたとしても，それより高いレイヤにおいては完全結合型に近いような密なネットワーク構成になっている場合が考えられる．さらに，ネットワークの冗長化構成を取っていた場合は，これの定数倍となる可能性がある．

このように，大企業や政府の中央省庁等の大規模な情報システム上で，複数のマルウェアが侵入したと想定する等，規模の大きい脅威トレースを行う場合には，シミュレーションにかかる計算の時間が非常に長くなるため，強力な 1 台の計算機で計算を進めるか，並列分散処理により複数の計算機で計算を分担して行う必要がある．並列分散処理により複数の計算機で計算を分担して行う場合，これにはいくつかの手法が考えられるが，ここでは並列分散処理のためのタスクスケジューリングの手法を用いることを考える．

ネットワークのモデルは，各ノードをタスク，各エッジをタスク間の依存関係と見なすと，互いに依存関係のあるタスク相互作用グラフ（Task Interaction Graph）として見ることができる．タスク相互作用グラフによって表されたネットワークのモデルは，実際に脅威トレースを行うための，動的な計算処理の前に，一度だけ静的な処理を行い，タスク依存グラフ（Task Flow Graph）に変換する．この処理はプログラミング言語のコンパイル処理と同等のもので，翻訳機械によって自動的にタスク相互作用グラフを解釈し，変換する．これにより，タスクを多くの計算機に適当に割り付ける際の流れが明確に定まり，スケジューラの負荷が少なくなるとともに，計算機の利用効率を高めることが可能である．ただし，情報システムの更改などによってネットワークの構成が変化した場合は，変化した後のネットワークの構成に合わせてモデルを再設定し，再度静的な処理によりタスク依存グラフを生成し直す必要がある．例を図 3 に示す．ネットワークのトポロジからタスク相互作用グラフを生成し，これからタスク依存グラフを生成する．

脅威トレースを行うための動的な処理を行うときは，静的な処理により生成したタスク依存グラフを参照して，タスクをいつどの計算機に割り当てるかを管理するタスクスケジューラが，管理下の計算機に動的にタスクを割り当て，計算を実行させる．

4. おわりに

筆者らは，組織の情報システムに対する「新しいタイプの攻撃」の成功を未然に防ぎ，情報システムの電子的破壊や内部情報の漏洩等を防ぐための脅威トレーサを実装するにあたっては，並列分散処理のためのタスクスケジューリングの手法を用いた，複数の計算機による効率的実行が必要であると考える．今回の予稿では，タスクスケジューリングの手法を情報システムの模擬するためにはどうすべきかということについて提案した．

脅威トレーサによる脅威トレースの自動化によって，机上で手計算により行われてきた従来のトレース作業に比べて，大規模でより実際的なネットワークに，複数のマルウェアが侵入している場合など，より複雑なトレースが可能になるほか，手計算では発生しがちな見落としや計算ミスを防ぎ，網羅的なトレースが可能になる．

タスクスケジューリングの手法により並列分散処理を行うにあたっては，タスクを様々な計算機に割り振って計算を行う必要がある．計算機にはいろいろな種類があり，たとえば，x86 互換の汎用プロセッサや，SIMD（Single Instruction Multiple Data）演算に特化した Cell/B.E.，これまではグラフィック演算のみに用いていた GPU を汎目的の計算に用いる GPGPU

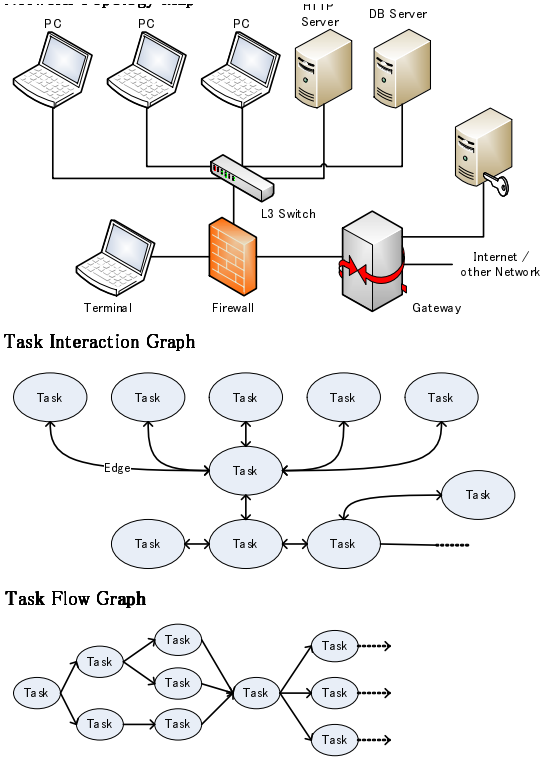

図 3 ネットワークトポロジ図とそれに対応するタスクグラフの例.
Fig. 3 An example of network topology, task interaction graph and task flow graph.

(General Purpose computing on Graphics Processing Unit)，メニーコアで多スレッドの高速並列演算を実現しているオラクルの SPARC T4 プロセッサなどがある．種類ごとに特性の違う計算機に，うまくタスクを割り振ることによって，タスクの実行効率を高めることができる．特に SIMD 演算をうまく用いて，脅威トレーサによる計算を SIMD 演算により実現することにより，飛躍的に計算の効率が高められると考えている．このことについての検討を行い，今後は並列分散処理化を施した実装を行う．

謝辞 本研究は，独立行政法人情報処理推進機構の支援および文部科学省科学研究費補助金（課題番号 21500039）および総務省の助成を受けている．

参 考 文 献

1) 独立行政法人情報処理推進機構:『新しいタイプの攻撃』の対策に向けた設計・運用ガイド, pp. 3–4, http://www.ipa.go.jp/security/vuln/documents/newattack.pdf (2011).
2) 独立行政法人情報処理推進機構:『新しいタイプの攻撃』の対策に向けた設計・運用ガイド, pp. 5–8, http://www.ipa.go.jp/security/vuln/documents/newattack.pdf (2011).
3) 独立行政法人情報処理推進機構: IPA テクニカルウォッチ:『新しいタイプの攻撃』に関するレポート, p. 6, http://www.ipa.go.jp/about/technicalwatch/pdf/101217report.pdf (2010).
4) 内閣官房情報セキュリティセンター: 平成 21 年度各専門分野情報共有スキームの連携性及び情報交換モデルの構築支援業務：リスク要件リファレンスモデル作業部会報告書, http://www.nisc.go.jp/inquiry/pdf/2-1_RM-model_Open.pdf (2009).
5) MITRE–Applying Systems Engineering and Advanced Technology to Critical National Problems, http://www.mitre.org/, 2011 年 11 月 17 日確認.
6) MITRE: MAEC–Malware Attribute Enumeration and Characterization, http://maec.mitre.org/index.html, 2011 年 11 月 17 日確認.
7) 小出 洋, 金岡 晃, 加藤 雅彦: 新しいタイプの脅威を分析するための情報システムとマルウェアを表現する DSL, 夏のプログラミング・シンポジウム 2011 報告集 (2012).
8) 金岡 晃, 原田 敏樹, 加藤 雅彦, 勝野 恭治, 岡本栄司: 安全なネットワークシステム設計のためのマルチレイヤネットワークモデルの提案と応用, 情報処理学会論文誌, Vol. 51, No. 9, pp. 1726–1735 (2010).
9) The Scala Programming Language, http://www.scala-lang.org/ 2011 年 11 月 18 日確認.
10) Odersky, M., Spoon, L. and Venners, B.: Programming in Scala, 2nd Edition (e-book), Artima, Inc., pp. 409–421 (2010).