# 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

小出 洋 [†1,†2] 金岡 晃 [†3] 加藤雅彦 [†4]

APT（Advanced Persistent Threats）と呼ばれる複雑で対応が難しい企業等の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．この種のサイバー攻撃の複雑な挙動は，セキュリティ分野のエキスパートにより事案毎に分析されているが，複雑で大規模な現在の情報システムにおいて，人間が網羅的かつ客観的で見落としの無い分析を効率的に行うのは難しい．そこで情報システムにおける脅威の挙動を計算機で網羅的にトレースして，エキスパートの専門知識と経験に基づく分析を補うことが考えられる．本発表では，本目的のために現在の情報システムとマルウェア等の脅威を表現する DSL について議論する．

# A Domain Specific Language to represent Information Systems and Malware for examining Advanced Persistent Threat

Hiroshi Koide ,[†1,†2] Akira Kanaoka [†3] and Masahiko Kato[†4]

It has been becoming a problem that Advanced Persistent Threats, APT, are cyberattacking information systems of enterprises and organizations. The activities of APT are examined by specialists but they are so complicated. However, it is very difficult to analyse the cyberattacks objectively and comprehensively on the latest realistic complicated information systems. So we propose a tracer which traces the activities of cyberattacks on the information systems comprehensively to help the examinations of specialists based on their expertise and experience. In this article, we discuss a domain specific language to represent information systems and malwares to examine APT.

## 1. はじめに

APT（Advanced Persistent Threats）などと呼ばれる情報システムに対する攻撃（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表記）を行うマルウェアに代表される脅威が問題となっている．本研究は，実際的な情報システムにその種の脅威が侵入したときにどのような活動が行われているのか，また脅威が行う攻撃を途中で阻止したり，情報漏洩を防いだりするにはどうすれば良いかを明らかにすることを目的としている．そのため，適当に抽象化されモデル化された脅威が情報システムに加えられたとき攻撃は情報システム上でどのような経路をとるかを，本研究で提案する「脅威トレーサ」によりシミュレーションすることで，実際に情報システム上で何が起きているのかを明らかにし，具体的な攻撃パターンに即した，現状の脅威に対抗可能なセキュリティ対策設計を可能化する．

近年，企業や政府等の組織は内部情報網の電子化が急速に進み，現在企業情報活動に不可欠である製品の設計情報，知的財産のリポジトリ，メールや文書のアーカイブ，顧客の個人情報等の機密性の高い情報が電子的に管理保管されている．それに伴い，情報システムに対する犯罪行為の目的や手法が変化しており，攻撃者個人が，技術力の誇示，愉快，あるいはクレジットカード情報等の個人情報の搾取を目的とした 1 台の PC や Web サーバを対象とした攻撃から，国際的犯罪組織や国家レベルの機関が企業や政府等の組織の機密性の高い情報の搾取を目的とした，情報システム全体を対象とした攻撃に変化してきている．

とりわけ，標的型メール攻撃等に代表される新しいタイプの攻撃は，国家のセキュリティやグローバル経済活動に関わる特定の組織にターゲットを絞って執拗

†1 独立行政法人情報処理推進機構
Information-technology Promotion Agency
†2 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
koide@ai.kyutech.ac.jp
†3 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
†4 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc.

かつ継続的な攻撃を行い，攻撃対象の情報システムの特徴や防御策に対応して攻撃方法を変化させ，密かに長期間の攻撃を行うことが多いとされている．未発見（ゼロディ）の脆弱性が利用されることも多く，従来のウィルス対策ソフトウェアや侵入検知システムによる対策だけでは不十分であり，新しいタイプの攻撃を行うマルウェアが情報システム内に侵入し，そこで何らかの活動が行われてしまうことは完全には防ぎきれないとされている．実際に企業組織の戦略上，重要な機密情報が狙われるという事例も起きている．事例を挙げると枚挙にいとまがないが，以下幾つかの重要なものを示しておく．

(1) マイクロソフト社の Internet Explorer に存在していたゼロディの脆弱性を利用してグーグル等の米国の特定の企業だけ狙ったオペレーション・オーロラと呼ばれる攻撃が行われた．実際に幾つかの企業の情報システムが被害を受けた（2010 年 1 月）．

(2) Stuxnet と呼ばれる複数のゼロディの脆弱性を利用した高度で複雑な標的型コンピュータワームを使って，特定の国の原子力プラントをターゲットにして，その制御システムに侵入して悪影響を与えるように仕組まれている．Stuxnet は 4000 弱の機能を持つ大規模ソフトウェアであり，国家レベルの組織が年単位の期間をかけて開発したと考えられている（2010 年 4 月）．

(3) 東日本大震災の直後から，それに乗じた震災情報の提供を装う標的型メール攻撃が極めて特定の分野の企業に対して集中的に行われた（2011 年 4 月〜9 月頃）．また，情報収集衛星光学 4 号機の打ち上げ前に開発に関わった企業に標的型メール攻撃が行われた（2011 年 9 月）．

(4) 衆議院の国会議員が利用する情報システムに対して標的型メール攻撃が行われ，国会議員を含む 4000 以上のアカウントが盗まれ，国防や外交上の機密事項を含むメール等の文書が少なくても 1ヶ月の間，外部の攻撃者が参照できる状況にあった（2011 年 10 月）．

企業団体活動に多大な影響を与える「新しいタイプの攻撃」に対する入口対策以外の対策，特に機密情報を外部に漏らさない出口対策の必要性が世界的に高まっている．例えば，内閣官房情報セキュリティセンターは，リスク要件リファレンスモデル[1] として調査研究をまとめており，対策の必要性をアピールしている．IPA は，新しいタイプの脅威に関する情報[2] や対策ガイド[3] をまとめて公開し，入口対策のみでは不十分であり，出口対策もあわせて行うことが有効であることを示している．

本研究ではマルウェアと情報システムをモデル化し，計算機によるマルウェアの挙動を追跡する手法を考えているが，マルウェアのモデル化の部分に関しては，MITRE の MAEC(Malware Attribute Enumeration and Characterization) をマルウェアのモデル化に関する既存研究として挙げることができる[5]．MAEC では，さまざまなマルウェアを分類整理してそれらの対策を立てるため，振る舞い，外観，攻撃パターンについての特徴づけを行う標準的な方法を提供することを目的としている．MITRE は，統一性があり明確なマルウェアのタイプ別の分類と系譜の追跡，正確なグループ化，記述ができると主張しており，実際に MAEC で分類したマルウェアの情報を標準的なフォーマットで記録した大規模なデータベースを構築しようとしている．

MAEC は亜種の多いマルウェアの整理と分類を行い，それらの対策を立てることが本来の目的である．そのために 1 台のホスト計算機内部におけるマルウェアの挙動を記述し．他のマルウェアと比較して，わずかな相違も詳細に記述して分類する能力を持つ．しかし，本研究の脅威トレースを行う目的で使用するには，複数のホスト計算機間で互いにネットワークを介して影響し合う挙動に関する十分な記述ができない．また，MAEC の計算機内部における詳細な記述能力が却ってマルウェアのモデル化という見地から冗長過ぎる結果をもたらすことになり，脅威トレースの実装の見地から効率的な実行の妨げとなっている．現に動的解析ツール等の MAEC を応用したツールは，まだ存在していない．

## 2. 脅威トレーサ

本研究で提案する脅威トレーサはマルウェアとそれが動作する情報システムを抽象化し，その挙動をシミュレーションにより予想する．このシミュレーションはノードの個数やマルウェアの複雑さに比例する．大企業や政府団体の実際的な規模の情報システムに適用したり，数千もの豊富な機能を持つ実際のマルウェアを模擬したりするには多くの計算時間を要するようになるため，並列分散化により計算時間を短縮化する必要がある．

現在までに IPA の「脅威と対策研究会」では，企業における情報システムとして，ルータやスイッチの他，ファイヤウォール，ゲートウェイ型ウィルスチェッカ，ロードバランサ，プロキシサーバを含む情報シス

```
object Sample extends MalwareSimulationLibrary {
  val pc1, pc2 = PCTerminal
  val router1 = Router
  val malware1 = Stuxnet
  val table1 = RoutingTable
  pc1 has a networkCard “192.168.0.2/24”
  pc1 opens port (80, 22)
  pc2 has a networkCard “192.168.1.2/24”
  pc2 opens port (80, 22)
  router1 has a networkCard “192.168.0.1/24”
  router1 has a networkCard “192.168.1.1/24”
  router1 has a routing table1
   pc1 connects to router1
  pc2 connects to router1
  pc1 is infectedWith malware1
  run
}
```

**図 1** DSL による情報システムの記述例.
Fig. 1 An example of network written by DSL.

テムの典型的なモデルを作成し，その上で標準的な標的型マルウェアの挙動について机上トレースを行なっている[1]．この机上トレースを人に代わって計算機で自動的に行うソフトウェアがわれわれが提案する脅威トレーサである．

本研究では，脅威トレーサの設計を行い，トレース対象のマルウェアとそのターゲットである企業や組織における大規模な情報システムを適切に抽象化して計算機でシミュレーションするための DSL を考察する．網羅性，再現性，トレースに要する時間について，人手による机上トレースとの比較を行い，脅威トレーサによる結果と人手による結果が補完的であるか等その有効性の検討を行う予定である．

## 3. 手　　法

### 3.1 情報システムのモデル化

金岡らが提案している NSQ モデル[4] を基本としたデータ構造を利用して，情報システム全体のスナップショットを表現する．その上でマルウェアの挙動および情報システムをステートフルで離散的な状態をとるオブジェクトとして扱い，離散イベントのシミュレーションにより変更を行い，情報システム内の機密情報が外部に持ち出される条件を見つける等の状態探索を行う．このとき，情報システムとマルウエアを適当な抽象度で抽象化して表現する DSL を定義し，脅威トレーサが直接扱えるインスタンスとして組み込む．例えば図 1 はルータを介して 2 台の PC を接続した単純な情報システムを DSL で表現した記述例である．

### 3.2 マルウェアのモデル化

脅威トレースを効率的に行うという本研究の目的に合わせたマルウェアのモデルを定義する．先ず共通部分が存在し，それを基本に多段攻撃を行う APT 的攻撃手法を表現する新しいモデルを作成し，DSL で記述しマルウェアのインスタンスに直接組み込み，シミュレーション時に実行することにより，マルウェアの挙動解析を行う．代表的なメソッドを表 3.2 に示す．Conficker と呼ばれる標的型ワームマルウェアの攻撃の挙動を机上トレース例[1] の一部を用い，代表的なメソッドの説明をする．

(1) 改竄された正規の Web サイトを参照すると悪性コードが実行され，標的型ワーム型マルウェアに感染する（`infectsTo`）．
(2) 感染したマルウェアが起動する（`invoke MalwareCore`）．http 通信がリダイレクションされ，攻撃者の C&C サーバに誘導する（`getsNetworkInfo`，`communicatesWith`）．
(3) ワーム本体が 2 段階に分けてダウンロードされる（`updatesMyselfBy`）．
(4) Web サイトを改竄（`attack`）したり，認証情報を盗聴（`getValuableInfo`）したりする挙動をとる．

### 3.3 離散シミュレーションと並列化

状態変数とリンク構造により抽象化されたマルウェアと情報システム全体の状態の離散イベントのシミュレーションを行い，機密情報を外部に持ち出す条件を探索する．このシミュレーションは時系列に沿った場合分けにより分岐し，ネットワークのノードとなる端末やルータ等の挙動を模擬する．

## 4. 実　　装

脅威トレーサの開発言語には Scala[6] を利用している．DSL の記述に向き，ネットワークとマルウェアの記述に利用する XML を強力に扱える．また情報システムのモデルのように多くの性質（例えば，経路制御，感染の可能等のマルウェアに対する性質，ソフトウェアが動作するか否か等）を実装して模擬する場合等，多くの性質を扱う場合，トレイトと呼ばれる機能毎に整理して開発して，後でミックス・イン合成を自動的に行い，ひとつにまとめることができる．最終的にソフトウェアの規模が大きくなった場合や並列分散化により複雑化しても対応可能であり，さらに Java の豊富なライブラリが利用でき，性能も良いという理由による．

### 4.1 情報システムの内部表現

脅威トレーサは以下の内部表現の情報システムの状態変数やノード間のリンク構造の状態について，離散イベント・シミュレーションを行う．

| メソッド名 | 引数 | 返値 | 説明 |
|---|---|---|---|
| infectsTo | 感染対象のノード | 結果 | PC に自身を感染させる |
| waitNormalComm | プロトコル | 最初に通信してきたノード | 通常通信を待つ |
| invokeMalwareCode | コード | 実行結果 | 悪性コードを実行する |
| communicatesWith | Malware, Message | 結果 | 別のマルウェアと通信する |
| updatesMyselfBy | アップデータ | 結果 | 自分自身をアップデートする |
| getsAttackableNode | なし | 攻撃可能ノードのリスト | 攻撃可能ノードのリストを得る |
| getsNetworkInfo | なし | ネットワーク関連の情報 | Proxy 情報等の情報を得る |
| getsValuableInfo | なし | 機密情報 | ノード上の機密情報を得る |
| attack | 攻撃対象, 攻撃方法 | 結果 | 攻撃を行う |

**表 1** マルウェアのモデルが持つ代表的メソッド.
Table 1 Principal methods which are implemented in the model of malwares.

**基本ノード:** NSQ モデル[4] に基づく，情報システムのノードにおける基本的な情報を保持，管理する．通し番号，識別ラベル，レイヤ，ノード種別，MAC アドレス，IP アドレス，TCP ポート，UDP ポート，サービス等についてのアクセッサメソッドを実装している．

**マルウェアに感染するノード:** マルウェアの感染を模擬する．感染したマルウェアを保持しておくリストを持つ．マルウェアに感染する（`getsInfectWith`），マルウェアを駆除する（`removeMalware`），マルウェアの動作を開始する（`invokeMalware`）のメソッドを実装している．

**ネットワークノード:** ノードに接続されているネットワークインターフェースのリスト，デフォルトルートの IP アドレス，ルーティングテーブル，ファイヤウォールの規則を保持し，ノードの IP ネットワークにおける挙動を模擬する．

**通信可能ノード:** 別の通信可能ノードとの到達可能性を確認し通信可能なときにメッセージを送る．`scala.actors.Actor` を継承したメッセージ通信で実現する．

### 4.2 離散イベント・シミュレーション

離散イベント・シミュレーションは，Scala の解説書のステートフルオブジェクトを扱っている箇所[7] で紹介されている汎用の離散イベント・シミュレータを基本にして，新たに実装を行うものである．全体の構成を図 2 に示す．

この本格的な実装を行う前に，間違えた思い込みや問題点がある場合は早めに明らかにする，想定する技術による実装が可能か，実行の様子について意識を共有し，マルウェアや情報システムを記述する DSL の設計や脅威トレースの設計と実装に役に立てることを目的とした試験実装までを行なっている．結果，この試験実装における離散イベント・シミュレータは情報システム上に存在する各マルウェアを順次動作させ，

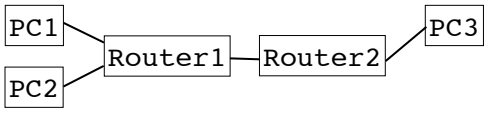


**図 3** テスト用の情報ネットワーク例.
Fig. 3 A sample of network for testing.

われわれが想定する方法によりマルウェアの実装が可能であることが分かった．

この試験実装に図 3 に示すテスト用の情報システムを与え，PC1 に簡単なマルウェアのモデルを感染させたとき，まず PC2 に感染が広がり，その後 PC3 に感染が広がる様子を図 4，5，6 に示す．

(1) 初期状態では情報システムは図 3 のようにルータと PC が接続されており，PC1 にマルウェアが感染している (図 4).

(2) PC1 のマルウェアは感染先を探索する．PC2 を発見して感染可能であるため，PC2 に感染する．実行ログの `print` の後はネットワークの状況を表しており，新たに PC2 がマルウェアに感染したことを示している (図 5).

(3) PC1 および PC2 のマルウェアは同時に活動し感染先を探索する．PC3 を発見して感染可能であるため，PC3 に感染する．実行ログによると，新たに PC3 がマルウェアに感染したことを示している (図 6).

## 5. おわりに

新しいタイプの脅威を分析するための脅威トレースを実装するにあたり，必須となる情報システムとマルウェアを表現する DSL について議論した．今後は，RAT(Remote Access Tools) を用いた標的型スパイ攻撃の特徴を捉えた有効な分析と対策をとれることを目的として，さらに考察を行っていく．また脅威トレー

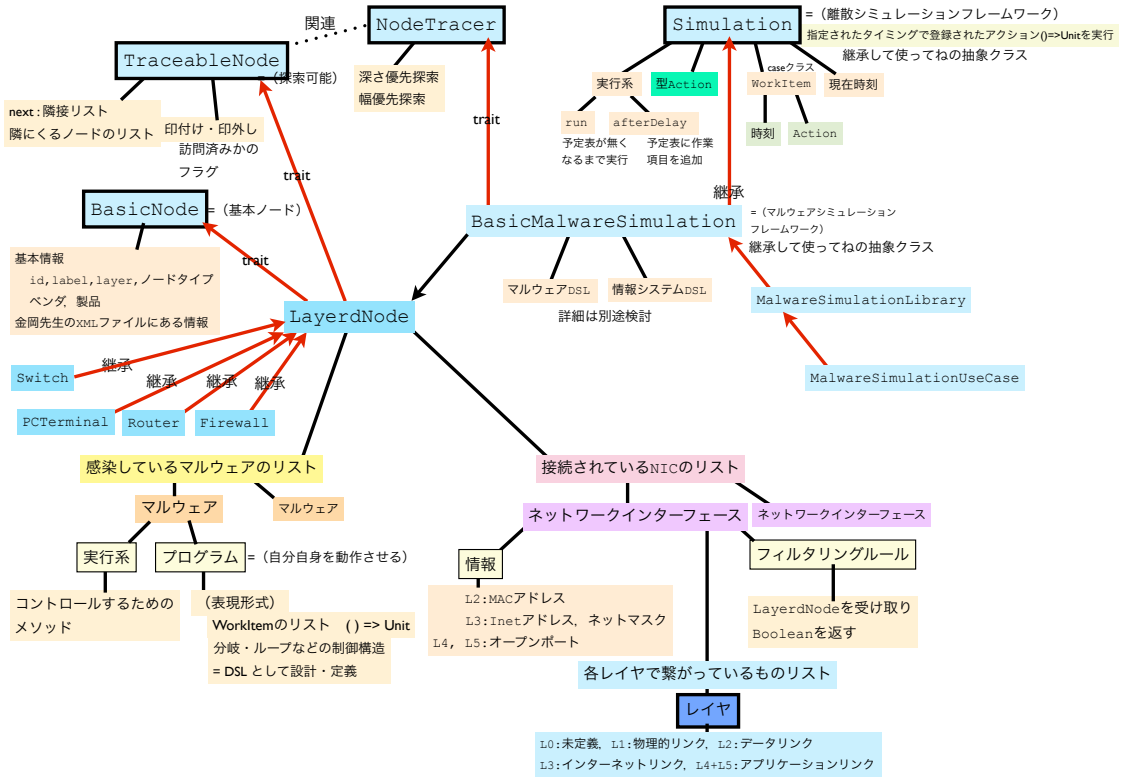


**図 2** 全体の構成.
Fig. 2 Configuration of all modules.

```
$ java -jar MalwareSimulator.jar
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
```

**図 4** 実行例（初期状態）.
Fig. 4 An example of testing (Initial state).

```
Malware(PCTerminal(pc1)):action
PCTerminal(pc1):send Router(router1)
Router(router1):send PCTerminal(pc1)
PCTerminal(pc1):send Router(router1)
Router(router1):send PCTerminal(pc2)
PCTerminal(pc2):send Router(router1)
Router(router1):send PCTerminal(pc1)
PCTerminal(pc2):call isInfectable true
print
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
Malware(PCTerminal(pc2))
```

**図 5** 実行例 (PC2 に感染).
Fig. 5 An example of testing (after PC2 is infected).

```
Malware(PCTerminal(pc1)):action
Malware(PCTerminal(pc2)):action
PCTerminal(pc2):send Router(router1)
  ＜＜中略＞＞
Router(router1):send PCTerminal(pc2)
PCTerminal(pc2):call isInfectable false
PCTerminal(pc1):send Router(router1)
PCTerminal(pc1):call isInfectable false
Router(router1):send Router(router2)
  ＜＜中略＞＞
Router(router1):send Router(router2)
PCTerminal(pc3):call isInfectable true
Router(router2):send PCTerminal(pc3)
PCTerminal(pc3):send Router(router2)
Router(router2):send Router(router1)
Router(router1):send PCTerminal(pc2)
PCTerminal(pc3):call isInfectable false
print
PCTerminal(pc1)
PCTerminal(pc2)
Router(router1)
PCTerminal(pc3)
Router(router2)
Malware(PCTerminal(pc1))
Malware(PCTerminal(pc2))
Malware(PCTerminal(pc3))
```

**図 6** 実行例 (PC3 に感染).
Fig. 6 An example of testing (after PC3 is infected).

スの設計と実装，並列分散化による高性能化を行なっていく方針である．

## 質疑応答

**C:** まだこれから明らかにしなければならないことが多々あるがユニークな研究であり，これから明らかになる結果が期待できる（山之上@鹿児島大）．

**Q:** 単にマルウェアの定性的な性質を調べるだけでは無く，動的にマルウェアそのものをシミュレーションする目的は何か（中山@NTT ソフトウェア）．

**A:** 動的にシミュレーションすることにより，マルウェアの本質的な性質が明らかになり，根本的な対策が可能になると考えている．感染した環境毎に影響はことなるが，それを現在行なっているのはプロフェッショナルによる人手の作業である．定性的なマルウェア解析だけでは，さまざまな環境に対する影響を 1 次影響，2 次影響など深く測るには不十分であり，コンピュータによる動的シミュレーションを繰り返して考察することが必要と考える．

## 謝　辞

本研究は独立行政法人情報処理通信機構の支援を受けている．

## 参考文献

1) 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, pp. 58 (2009).

2) 独立行政法人情報処理推進機構：「新しいタイプの攻撃」に関するレポート，http://www.ipa.go.jp/about/technicalwatch/20101217.html (2010).

3) 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，http://www.ipa.go.jp/security/vuln/newattack.html (2011).

4) 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).

5) MITRE: Malware Attribute Enumeratation and Characterization, http://maec.mitre.org (2011).

6) The Scala Programming Language, www.scala-lang.org (2008).

7) Odersky, M., Spoon, L. and Venners, B.: Programming in Scala, 2nd Edition (e-book), Artima, Inc. pp. 409-421 (2010).