サイバー攻撃対策のための観測記述データ表記に関する検討

寺田真敏　岩本一樹　遠藤基
松坂志　　小林偉昭

マルウェアの攻撃手法の多様化と巧妙化は進んでおり、活動形態にも大きな変化がみられるだけではなく、個々のフェーズ毎でも、機能面や実装面での変化が見られる。本研究の目的は、このような機能面や実装面での変化を記録として残しつつ、その変化点を捉えて、効果的な対策につなげることにある。本稿では、2000 年代前半に流布したメール型ワームの実装面での変遷について事例調査の結果を示す。次に、マルウェアの外部通信に着目して、変遷過程を記録する目的、記録情報について検討した結果を報告する。

Feasibility study of Observable Expression for Cyber Attack Countermeasure

Masato Terada, Kazuki Iwamoto, Motoi Endo
Nozomi Matsuzaka and Hideaki Kobayashi

Malware activities are progressing and changing for long term and short term, especially for the aspects of the function and implementation. For these progressing and changing, we should track them and keep up-to-date and reviewing for our information system security. In this paper, firstly we will describe implementation transition of email worm. Secondly, we will introduce our tracking approach for the aspects of the functional and implementation changing of outbound traffic of malware.

1. はじめに

マルウェアの攻撃手法の多様化と巧妙化は進んでおり、活動形態にも大きな変化がみられる。1999 年頃からメールを介したマルウェアの受動型感染が始まった。2001 年頃からはネットワーク型ワーム、2004 年頃からは遠隔操作可能なボットが流布した。その感染形態は、感染対象のホストに対してマルウェア自身が攻撃コードを送信する能動型感染が主流であった。2008 年頃からは、ブラウザが利用するプラグインやアプリケーションの脆弱性を利用して、マルウェアをダウンロードして実行する攻撃手法、ドライブバイダウンロード(drive-by-download)を用いた Web 感染型マルウェアが流布した。2011 年に入ると、メールと遠隔操作ツール(Remote Access Trojan/Remote Administration Tool)とを組合わせた組織内ネットワークへの侵害活動が台頭してきている。

このような大きな活動形態の変化と共に、個々のフェーズ毎でも、機能面や実装面での変化が見られる。例えば、マルウェアによる外部通信(イントラネットからインターネットに向けての通信)の特徴として、2008 年の文献[1]によれば、80/tcp、443/tcp を使用するが、いずれも独自プロトコルであるとしている。2012 年の文献[2]によれば、80/tcp のうち、93%が HTTP を、残り 7%が独自プロトコル、443/tcp の場合には HTTP と独自プロトコルが半々となっている。

本研究の目的は、このような機能面や実装面での変化を記録として残しつつ、その変化点を捉えて、効果的な対策につなげることにある。そこで、本稿では、2000 年代前半に流布したメール型ワームの実装面での変遷について事例調査の結果を示す。次に、2011 年に注目された出口対策を対象に変遷過程を記録する目的、記録情報について検討した結果を報告する。

(独)情報処理推進機構
Information-technology Promotion Agency, Japan

2. 関連研究

2.1 サイバー攻撃対策モデル

攻撃対象となる組織に合う手法を選択し(標的型)、組織内ネットワークを活動基点とした(潜伏型)侵害活動といわれているAPT(Advanced Persistent Threat)については、その進行段階をモデル化し対策を検討する試みが行われている。文献[3]では、Exploitation Life Cycle と呼ぶ Reconnaissance(偵察)、Initial Intrusion into the network(侵入)、Establish a Backdoor into the network(遠隔制御)、Obtain user Credentials(権限取得)、Install Various utilities(インストール)、Privilege escalation/Lateral Movement/Data Exfiltration(実行)、Maintain Persistence(潜伏)から成る7段階を提案している。文献[4]では、より対策視点でモデル化するため、米国空軍の軍事コンセプトである Kill Chain(F2T2EA) をサイバーに応用し、Reconnaissance(偵察)、Weaponization(武器化)、Delivery(配送)、Exploitation(攻撃)、Installation(インストール)、Command and Control(C2)(遠隔制御)、Actions on Objectives(実行)の 7 段階から成る Cyber Kill Chain モデルを提案している。また、初期段階から対策として、配送段階での検知、武器化段階以前の分析と、攻撃者の意図、攻撃者のパターン、行動、TTP(Tactics, Techniques and Procedures：戦術、技術及び手順)を明らかにする攻撃活動分析(Campaign Analysis)の必要性を示している(図 1)。

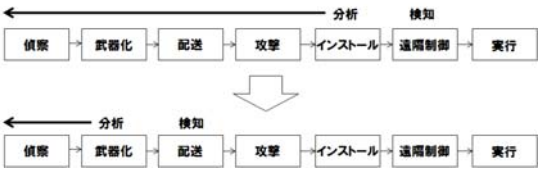

図 1：初期段階からの対策への移行

2.2 サイバーセキュリティ情報交換仕様

機能面や実装面での変遷を記録として残すという側面から、関連するサイバーセキュリティ情報交換仕様の状況について述べる。

(1) X.cybex(Cybersecurity Information Exchange Framework) [5]

ITU-T において、脆弱性対策情報のフォーマット、番号体系などの技術仕様を規格化するサイバーセキュリティ情報交換フレームワーク X.cybex(X.1500)の標準化が進められている。X.cybex は、Making Security Measurable の仕様群を中心に構成されている。

(2) Making Security Measurable [6]

情報セキュリティに関連する共通識別子、共通仕様を整備することにより、情報セキュリティ対策全般に関わる処理の機械化も促進されることになる。米 MITRE 社では、2007 年頃から共通識別子、共通仕様、リポジトリの視点から”Making Security Measurable”と呼ぶ活動を推進している。米連邦政府で使われているプラットフォームを自動チェックするための仕様群 SCAP(Security Content Automation Protocol)を構成する仕様群の他に、攻撃パターンの識別を共通化する CAPEC(Common Attack Pattern Enumeration and Classification)、マルウェアの動作、痕跡などの属性を記述する MAEC(Malware Attribute Enumeration and Characterization)、イベントの識別を共通化する CEE(Common Event Expression)などがある。

(3) CyBOX(Cyber Observable eXpression) [7]

米 MITRE 社が開発したサイバー攻撃観測記述言語 CybOX は、Mandiant の OpenIOC の仕様を踏まえた、サイバー攻撃活動での観測事象を記述するための XML 仕様である。2009 年 9 月に CAPEC の延長で検討が開始され、2010 年 12 月に Mandiant OpenIOC との連携、その後、CEE、MAEC Ver2.0 との連携が始まり、2012 年 4 月に Ver1.0 がリリースされた。Cyber Kill Chain モデルに基づく対策活動での活用が検討されている。

3. メール型ワームを対象とした事例調査

本章では、機能面や実装面での変化の事例として、2000 年代前半に流布したメール型ワームのメール送信方法とメールアドレス収集方法を取り上げる。

3.1 マルウェアのメール送信方法

2000 年代前半のマルウェアは、メールを介し

て自己複製することにより流布するワーム(メール型ワーム)が主流であった。本節では、メール型ワームが自己複製のために使用したメール送信方法の変遷について述べる。

(1) 実装の分類

メール送信方法は大きく WSOCK32.DLL に感染してメール送信する、MAPI を利用する、SMTP を利用するタイプに分かれる(表 1)。

表 1：メール送信方法に関する実装の分類

(a)WSOCK32.DLL		
(b)MAPI		
(c)SMTP	特定	オープンリレーSMTP サーバ
	送信側	Outlook Express の SMTP サーバ
		その他のメールクライアントの SMTP サーバ
	受信側	メールアドレスドメイン部に mail.や smtp.を付加
		DNS の MX レコード

(a) WSOCK32.DLL

WSOCK32.DLL の connet や send などの API にマルウェアのコードを挿入し、通信を監視する方法。マルウェアはメールクライアントが確立したコネクションを利用して、通常のメール送信に重畳してマルウェア添付メールを送信する。メールの送信タイミングはメールクライアントの動作に依存する。

(b) MAPI

Windows のメール送信用 API を利用する方法。マルウェア自身が SMTP プロトコル処理を実装する必要はないが、Outlook や Outlook Express がインストールされていることが前提となる。

(c) SMTP

マルウェア自身が SMTP プロトコル処理を実装する方法。特定の SMTP サーバを利用する方法、送信側の SMTP サーバを利用する方法、受信側の SMTP サーバを利用する方法に分かれる。

- オープンリレーSMTP サーバ
  事前登録されたオープンリレーの SMTP サーバを利用する。
- Outlook Express の SMTP サーバ
  レジストリの HKEY_CURRENT_USER ¥Software ¥Microsoft¥Internet Account Manage 配下に設定された Outlook Express の SMTP サーバを利用する。
- その他のメールクライアントの SMTP サーバ
  Outlook Express 以外のメールクライアント(Netscape や Eudora など)で設定された SMTP サーバを利用する。
- メールアドレスドメイン部に mail.や smtp.を付加
  test@example.jp に送信するときに、SMTP サーバとして mail.example.jp や smtp.example.jp を選択する。
- DNS の MX レコード
  DNS の MX レコードから SMTP サーバを特定する。

(2) 実装の変遷

1999 年 2 月の W95/Ska～2006 年 12 月の W32/Nuwar まで、118 検体を調査対象とした。調査結果として、メール送信方法の年毎の実装状況(図 2)、主となるメール送信方法に失敗した際に代替手段として選択されたメール送信方法(図 3)を示す。なお、Mydoom、Bagle、Netsky などの多数存在する亜種も積算の対象としている。

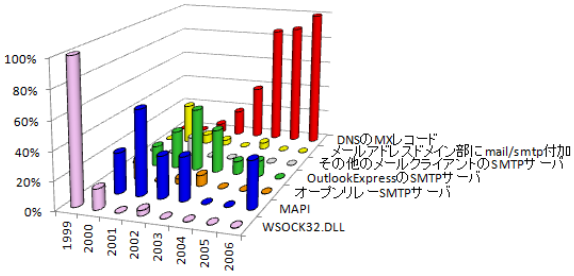

図 2：メール送信方法の実装状況

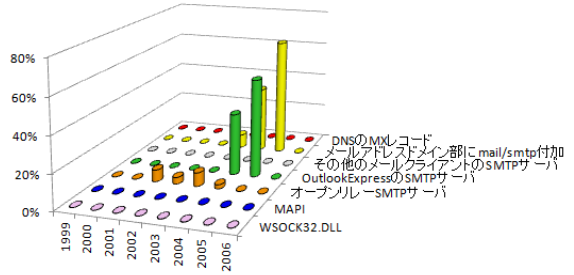

図 3：代替手段として実装された
メール送信方法

(3) 考察

WSOCK32.DLL を書き換える方法は実装面の難度から、模倣されることは少なかったと思われる。また、MAPI によるメール送信方法、オープンリレーの SMTP サーバを使用する方法は、MAPI へのマルウェア対策、該当するオープンリレーの SMTP サーバへの対策で無効化されてしまうことから、継続的に使用されることがなかったと考えられる。最終的に、DNS サーバの MX を参照してメール送信する方法が主流となるが、その背景には、W32/Mydoom.A のソースコードが公開されたことで実装方法の収束が起きたと考えられる。なお、W32/Mydoom.A を模したマルウェアとして、W32/Bagle、W32/Netsky、W32/Mytob がある。この後メール型ワームが衰退した理由としては、OP25B(Outbound Port 25 Blocking)によるメール送信の制限、メールサーバでの対策が進んだことなどが原因だと思われる。

3.2 マルウェアのメールアドレス収集方法

本節では、メール型ワームが実装した送信先メールアドレスの収集方法の変遷について述べる。

(1) 実装の分類

メールアドレスの収集方法は大きく WSOCK32.DLL に感染して収集する、MAPI を利用する、ファイル探索、その他のデータベースを利用するタイプに分かれる(表 2)。

表 2：メールアドレスの収集方法に関する実装の分類

(a)WSOCK32.DLL	
(b)MAPI	受信トレイ
	アドレス帳
(c)ファイル探索	Outlook Express のアドレス帳
	アドレス帳らしきファイル
	ファイル全般
(d)その他のデータベース	

(a) WSOCK32.DLL

WSOCK32.DLL の connet や send などの API にマルウェアのコードを挿入し、通信を監視する方法。W32/Hybris は送受信されるデータからメールアドレスらしきものを収集していた。

(b).MAPI

Windows のメール送信用 API を利用する方法。マルウェア自身がメールアドレスの収集のために、ファイルの探索や解析をする必要はないが、Outlook や Outlook Express がインストールされていることが前提となる。

- 受信トレイ
  受信トレイに格納されているメールに返信することでメールアドレスの取得を代用する方法。未読メールに限り返信するタイプと、すべての受信メールに返信するタイプがある。
- アドレス帳
  アドレス帳からメールアドレスを取得する方法。

(c) ファイル探索

- Outlook Express のアドレス帳
  レジストリの HKEY_CURRENT_USER ¥SOFTWARE¥SOFTWARE¥Microsoft¥WAB ¥WAB4¥Wab File Name から Outlook Express のアドレス帳の位置を特定し、ファイルを読み込みメールアドレスを取得する方法。この場合、アドレス帳の格納形式を知っている必要がある。
- アドレス帳らしきファイル
  アドレス帳らしきファイル検索し、メールアドレスを取得する方法。Outlook Express だけではなく、Netscape や Eudora などに対応していた事例もある。
- ファイル全般
  ファイル全般を検索し、ファイルの中からメールアドレスらしき文字列を取得する方法。「mailto:」などの文字列を探す場合もあるが、ファイルの格納形式に関係なく、「@」とその前後の文字列を取得するのが一般的である。このため、誤ったメールアドレスを取得する可能性も高い。

(d) その他のデータベース

ICQ や MSN メッセンジャーのコンポーネントを使用してメールアドレスを取得する方法。MAPI 同様に簡単で確実にメールアドレスを取得できるが、対象となるソフトウェアがインス

トールされていることが前提となる。

(2) 実装の変遷

1999 年 2 月の W95/Ska～2006 年 12 月の W32/Nuwar まで、118 検体を調査対象とした。調査結果として、メールアドレス収集方法の年毎の実装状況を示す(図 4)。なお、Mydoom、Bagle、Netsky などの多数存在する亜種も積算の対象としている。

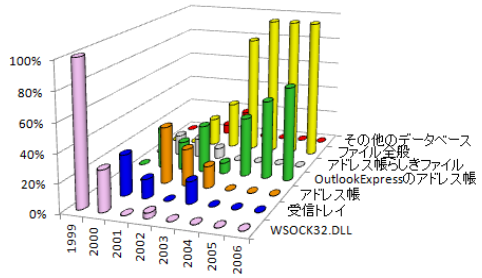

図 4：メールアドレス収集方法の実装状況

(3) 考察

ファイル探索によるメールアドレスの収集では、当初 Outlook Express のアドレス帳を読み込むのが主流であった。しかし、メールアドレスが記載されていると思われるファイルから、メールアドレスらしき文字列を収集するという方法へと変化した。この背景には、誤った文字列を取得してしまう可能性もよりも、過去にやり取りをしたメールアドレス、Web サイトに掲載されたメールアドレスなど、多岐に渡ってメールを送信できる側面が選択されたと考えられる。また、ICQ や MSN メッセンジャーのコンポーネントを利用する試みもあったが、主流とはならず消えた背景には、ファイル検索によるメールアドレス収集で間にあったからと考えられる。

4. マルウェアの外部通信に着目した検討

本章では、マルウェアの外部通信を対象に変遷過程を記録する目的、記録対象の候補、記録するためのフォーマットについて検討した結果を述べる。

(1) 目的

マルウェアによる外部通信(イントラネットからインターネットに向けての通信)において利用されるプロトコルは、2011 年と 2012 年では違いが見られる(図 5、図 6)[a]。この報告によれば、構成面ではプロキシに対応した通信、プロトコル面では独自プロトコルよりは HTTP プロトコルの選択、さらに、ポート番号は一般的な 80/tcp、443/tcp が選択される傾向が見られる。

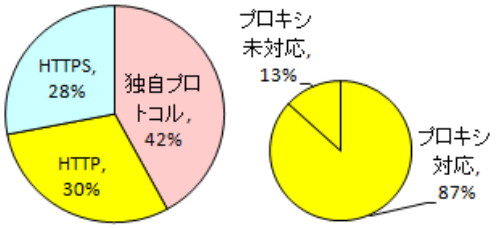

図 5：外部通信プロトコル(2011 年)

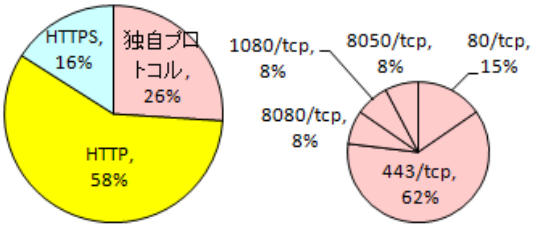

図 6：外部通信プロトコル(2012 年)

上述の調査結果を、次の 3 つのネットワーク構成に当てはめてみると、ルータ型 FW の構成はマルウェアの外部通信に対して耐性が低いことになる。

- ルータ型 FW
  80/tcp(HTTP、独自プロトコル)、443/tcp(HTTPS、独自プロトコル)の通信が発生
- 非認証型プロキシ
  プロキシでの CONNECT 接続可能なポート番号を 443/tcp、563/tcp に限定した場合、80/tcp(HTTP)、443/tcp(HTTPS、独自プロトコル)の通信が発生
- 認証型プロキシ
  非認証型プロキシに Basic 認証を追加した場合、80/tcp(HTTP)、443/tcp(HTTPS、独自

a) 出典：トレンドマイクロ：2011 年は 2011 年 4 月～10 月に、国内で収集された標的型攻撃メールに添付されていたと思われるマルウェア 50 検体のバックドア通信を対象。2012 年は文献[2]を参照。

プロトコル)の通信が発生

今後、構成面、プロトコル面の双方から、どのような通信形態が増えてきたのかを捉えていくことで、攻撃者側の機能面や実装面での変遷を明らかにすると共に、変化点をトリガとして、対策見直しの機会を持つ必要がある。なお、ここでは、マルウェアの外部通信として、HTTP、HTTPS、独自プロトコルに絞っているが、DNS、メールなども対象となる。

(2) 記録情報

マルウェアの外部通信に着目した場合、ポート番号を起点として、構成面とプロトコル面の 2 つの視点から検討すると、攻撃者側の実装面での変遷を記録にあたって必要となる情報は図 7 のようになる。443/tcp の場合には、HTTP を HTTPS と置き換えれば同じ構成になると考える。

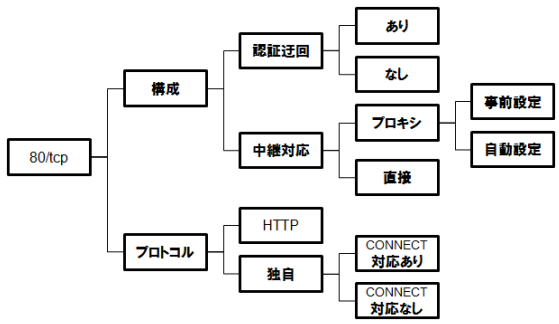

図 7：80/tcp を対象にした場合の記録情報

(3) 記録情報のフォーマット

攻撃者側の機能面や実装面での変遷を明らかにするにあたっては、変化点を捉えるために 3 段階のレベル分けの考え方を導入する(表 3)。

表 3：記録情報のレベル分け

レベル	概要	例
1	対策の一次情報として利用できるデータ	マルウェアのハッシュ値、接続先 IP アドレスなど
2	レベル 1 の補足情報や対策の阻害要因を示すデータなど	プロキシ対応、認証迂回機能など
3	今後、対策の阻害要因になるかもしれない情報など	マルウェア自身の防衛機能など

レベル 3 は、レベル 1 や 2 として記録していく必要があるデータを抽出することを目的としている。レベル 2 は、マルウェアの外部通信を対象とした場合、図 7 の記録情報や表 4 に示す Windows API の使い方が該当する。

表 4：Windows API の使い方

項目	概要
WSA 系(WSASelect や WSAConnect、WSARecv などの API)	Windows における最もネイティブな API
InternetOpen 系(InternetOpen や InternetOpenUrl などの API)	HTTP や FTP などのプロトコルを意識することなく通信できる。HTTP ヘッダなどは API 提供側で作成し、OS 設定値のプロキシの利用も可能
URLDownloadToFile 系(ファイル取得用の API)	Downloader やシェルコードが使う場合が多く、パラメータとして ID や OS のバージョンなどを示す少量のデータを送ることに利用できる。プロキシは自動的に OS 設定値を利用する。

レベル 1 については、サイバー攻撃観測記述言語 CyBOX を利用できる。ここでは、MWS 2012 Datasets の D3M 2012[8]で提供されている SHA1: 545e261b3b00d116 の外部通信を適用事例として示す。SHA1: 545e261b3b00d116 (20110725.exe) は、図 8 に示すように、xxx1wahaha.cn から 20110725.exe としてダウンロードされ、xxx2wahaha.cn への接続を試みる実行プログラムである。

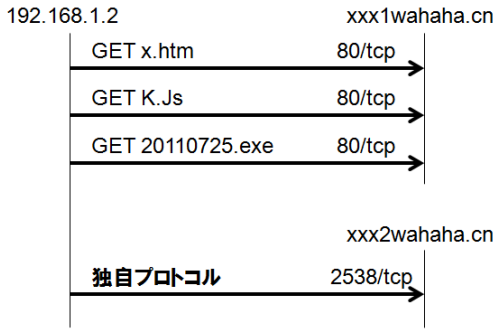

図 8：SHA1: 545e261b3b00d116 のダウンロードから外部通信までの動作

```
<cybox:Observable>
<cybox:Stateful_Measure>
  <cybox:Description>
  <common:Text>MWS 2012 Datasets - D3M 2012 - SHA1: 545e261b3b00d116 の特徴量 ファイルサイズ、ハッシュ値 接続先 URL、接続先 IP アドレス、ポート番号、プロトコル(TCP/UDP)</common:Text>
  </cybox:Description>
  <cybox:Object id="cybox:mws2012.d3m.545e" type="File">
    <cybox:Defined_Object xsi:type="FileObj:FileObjectType">
      <FileObj:File_Name datatype="String">20110725.exe</FileObj:File_Name>
      <FileObj:Size_In_Bytes datatype="UnsignedLong">16896</FileObj:Size_In_Bytes>
      <FileObj:Hashes>
      <common:Hash>
        <common:Type datatype="String">SHA1</common:Type>
        <common:Simple_Hash_Value condition="Equals"
        datatype="hexBinary">545e261b3b00d116</common:Simple_Hash_Value>
      </common:Hash>
      </FileObj:Hashes>
    </cybox:Defined_Object>
    <cybox:Related_Objects>
      <cybox:Related_Object idref="cybox:mws2012.d3m.file.exe.site" type="URI"
      relationship="Downloaded_From" />
      <cybox:Related_Object idref="cybox:mws2012.d3m.545e.con2" type="URI"
      relationship="Connected_To" />
      <cybox:Related_Object idref="cybox:mws2012.d3m.545e.resolvedto" type="IP Address"
      relationship="Resolved_To" />
      <cybox:Related_Object idref="cybox:mws2012.d3m.545e.con2protocol" type="Port"
      relationship="Connected_To" />
      <cybox:Related_Object idref="cybox:mws2012.d3m.545e.con2port" type="Port"
      relationship="Connected_To" />
    </cybox:Related_Objects>
  </cybox:Object>
</cybox:Stateful_Measure>
</cybox:Observable>

<cybox:Observable>
<cybox:Stateful_Measure>
  <cybox:Object id="cybox:mws2012.d3m.545e.con2" type="URI">
    <cybox:Defined_Object xsi:type="URIObj:URIObjectType" type="URL">
    <URIObj:Value datatype="AnyURI" condition="Equals">xxx2wahaha.cn</URIObj:Value>
    </cybox:Defined_Object>
  </cybox:Object>
</cybox:Stateful_Measure>
</cybox:Observable>

<cybox:Observable>
<cybox:Stateful_Measure>
  <cybox:Object id="cybox:mws2012.d3m.545e.resolvedto" type="IP Address">
    <cybox:Defined_Object xsi:type="AddrObj:AddressObjectType" category="ipv4-addr">
    <AddrObj:Address_Value datatype="String"
    condition="Equals">xxx.108.235.94</AddrObj:Address_Value>
    </cybox:Defined_Object>
  </cybox:Object>
</cybox:Stateful_Measure>
</cybox:Observable>

<cybox:Observable>
<cybox:Stateful_Measure>
  <cybox:Object id="cybox:mws2012.d3m.545e.con2port" type="Port">
    <cybox:Defined_Object xsi:type="PortObj:PortObjectType">
    <PortObj:Port_Value datatype="PositiveInteger" condition="Equals">2583</PortObj:Port_Value>
    <PortObj:Layer4_Protocol datatype="String" condition="Equals">TCP</PortObj:Layer4_Protocol>
    </cybox:Defined_Object>
  </cybox:Object>
</cybox:Stateful_Measure>
</cybox:Observable>
```

図 9：CyBOX を用いた SHA1: 545e261b3b00d116 の特徴量記述

図 8 の SHA1: 545e261b3b00d116 が xxx2wahaha.cn への接続を試みる部分を、マルウェアの外部通信として CyBOX で記述すると、図 9 のようになる。

5. おわりに

本稿では、マルウェアの機能面や実装面での変化を記録として残しつつ、その変化点を捉えて、効果的な対策につなげるためのアプローチについて示した。

まず、実装面での変遷として、2000 年代前半に流布したメール型ワームの事例調査を通して、メール送信方法とメールアドレス収集方法のいずれにおいても、実装面での変化点があることを示した。次に、マルウェアの外部通信に着目し、対策にあたっては構成面、プロトコル面の双方から、どのような通信形態が増えてきたのかを捉えていくことと、変化として記録すべき情報を示した。さらに、記録にあたっては、3 段階のレベル分けの考え方と共に、対策の一次情報として利用できるデータについては、サイバー攻撃観測記述言語 CyBOX による記述事例を示した。

今後の課題は、3 段階のレベル分けの実現方法として、特にレベル 2、3 の記録フォーマットなどの記録方法を具体化すると共に、変化点を捉えるための事例収集が挙げられる。さらに、その延長として、サイバー攻撃の対策のためのモデル化、攻撃活動分析、対策処理基盤の整備などを検討していきたいと考えている。

参考文献

1) IPA：近年の標的型攻撃に関する調査研究－調査報告書－(2008 年 3 月), http://www.ipa.go.jp/security/fy19/reports/sequential/

2) トレンドマイクロ：2012 年上半期国内における持続的標的型攻撃の傾向(2012 年 8 月)

3) MANDIANT：M-Trends レポート(2010 年 1 月)

4) Eric M. Hutchins, et.al.：Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains (2011 年 3 月)

5) Cybersecurity Information Exchange techniques (CYBEX), http://www.itu.int/en/ITU-T/studygroups/com17/Pages/cybex.aspx

6) MITRE: Making Security Measurable, http://measurablesecurity.mitre.org/

7) CyBOX, http://cybox.mitre.org/

8) MWS2012 実行委員会, 研究用データセット MWS 2012 Datasets について, http://www.iwsec.org/mws/2012/about.html#datasets