サンドボックス解析結果に基づくURLブラックリスト生成についての一検討

畑田 充弘† 田中 恭之† 稲積 孝紀†

†NTT コミュニケーションズ株式会社
108-8118 東京都港区芝浦 3-4-1 グランパークタワー16F
{m.hatada, yasuyuki.tanaka, t.inazumi}@ntt.com

あらまし マルウェア感染後の早期検知手法としては，システムログや通信トラフィックによる異常検知があるが，本稿では企業等におけるインターネット通信の代表例であるHTTP通信に着目し，出口対策としてのURLブラックリスト生成方式を提案する．サンドボックス解析結果から得られるマルウェアの通信先URLを用いるが，正常通信によるURLと区別が難しいものもあり，有効な手法としてテイント解析の研究が進んでいる．しかしながら，解析コストなどの課題もあるため，サンドボックス解析結果をもとにシステム情報やユーザ情報の読み取りを条件とした簡易なURLブラックリスト生成方式について，事例を示すとともに課題を考察する．

A Study on Light-weight URL Blacklist Generation based on Sandbox Analysis

Mitsuhiro Hatada† Yasuyuki Tanaka† Takanori Inazumi†

†NTT Communications Corporation
Gran Park Tower 16F, 3-4-1, Shibaura, Minato-ku, Tokyo 108-8118, JAPAN
{m.hatada, yasuyuki.tanaka, t.inazumi}@ntt.com

Abstract To detect the malware infection in internal network, not only anomaly detection but URL blacklist is also effective method that is focusing on HTTP traffic as typical example of the Internet traffic in organizations. Although URL blacklist can be extracted from the result of the malware analysis by sandbox, some URL are difficult to distinguish them from normal traffic such as checking the Internet reachability to major site. Taint analysis can be effective approach for identifying the malicious URL such as C&C or information leakage, but faces some technical challenges. In this paper, we present a novel approach of URL blacklist generation based on whether the malware read the system information or user credentials or not. We analyze and discuss the cases of our light-weight approach.

1 はじめに

AV Comparatives によると、主要なマルウェア対策ソフトが 90%以上の検知率で悪意のある Web サイトへのアクセスを検知・防御できていると報告がある[1]が，テストデータは独自に収集した URL（実行ファイルへの直接リンクや Drive-by download 含む）や手動で検索した URL，研究者から提供された URL 等をもとにしており，その網羅性については不明である．

一方で，感染を防ぐための「入口対策」の限界として，マルウェアの多様化，ゼロデイ脆弱性の悪用，脆弱性を持つソフトウェアの多様化，通常の通信に紛れたバックドア通信が挙げられており，「出口対策」の重要性が取り上げられている．

マルウェアへの感染を防止する入口対策には，ファイアウォールによるIPアドレスやポート番号，アプリケーション識別によるアクセス制御，侵入検知／防御システムによるシグネチャを用いた攻撃検知等がある．IP アドレスや URL のブラックリストを用いたアクセス制御が可能[3]であり，マルウェアへの感染を検知する出口対策としても利用できる．また Web プロキシでの URL フィルタも行われている[4]．

入口対策用 URL ブラックリストを生成する代表的な手法として，クライアント型ハニーポットによる悪性サイトの検知[5]がある．Web サイトに設置された exploit やマルウェアを検知するという明確な根拠を得ることができる．一方で，出口対策用 URL ブラックリストを生成するためには，静的解析によりマルウェアがアクセスする URL を抽出したり，動的解析により実際にマルウェアがアクセスする URL を記録したりする方式が考えられる．しかしながら，マルウェアがアクセスする URL にはインターネットの接続確認やグローバル IP アドレスの確認等も含まれているため，C&C 通信，マルウェアのダウンロード，外部への情報送信など不正なアクセス先 URL を区別する必要がある．

本稿では，企業等におけるインターネット通信の代表例である HTTP 通信と，適用箇所が広い URL ブラックリストに着目し，特に出口対策としての URL ブラックリスト生成方式を提案する．

2 関連研究

動的解析で得られるマルウェアの通信先には，①インターネット接続確認，②C&C 通信，③マルウェアのダウンロード，④スパムメール送信，⑤感染拡大のための攻撃，⑥DoS，⑦外部への情報送信、などが挙げられる．このうち，①はメジャーな Web サイトへのアクセスによって，インターネットへの接続確認や，IP アドレス確認サイトでのグローバル IP アドレスの確認等，正常な通信との区別が困難である．④は組織等のセキュリティ・ポリシーにも依存するが，Web メールによるスパムメール送信なども大量送信でなければ正常な通信との区別が困難である．⑤の攻撃先は近隣 IP アドレス帯や②によって指定される場合（⑥も同様）あるが，その対象が不正サイトであるとは言い難い．このようなトラフィックの特徴に基づく感染検知手法としては，ヘッダによるもの[6,7]，ペイロードによるもの[8]などがあるが，URL ブラックリストを生成するものではない．

アクセス先が不正サイトといえる，つまり URL ブラックリストとして利用可能な②③⑦について，②はシステムコールのビヘイビアグラフとデータフロー解析によって，外部への送信データや受信データに基づいて C&C 通信を判定するもの[9]がある．③はダウンロードしたファイルがマルウェアかどうかによって判定することができる．⑦はテイント解析[10, 11]によって外部へ感染ホストのシステムやユーザ固有の情報などの送信先を判定するものもある．特に②や⑦については，ビヘイビアグラフの網羅性や，テイントの伝搬漏れ・誤伝搬などとともに，解析コストが大きいことも課題である．

3 提案方式

出口対策用 URL ブラックリストを生成するために，関連研究等による厳密な不正サイトの判別も有効ではあるが，近年では多様な動的解析環境（サンドボックス[12]）が利用できることから，簡易な方式を提案する．

3.1 処理概要

図 1 に処理概要を記述する．マルウェアをサンドボックスで解析し，解析結果（レポート）を得

る. 予め定義したチェック対象と一致した場合，当該マルウェアのアクセス先 URL を解析結果から抽出する. メジャーサイト等のホワイトリストに該当するものを除き，URL ブラックリストを得る. 以下，各処理について述べる.

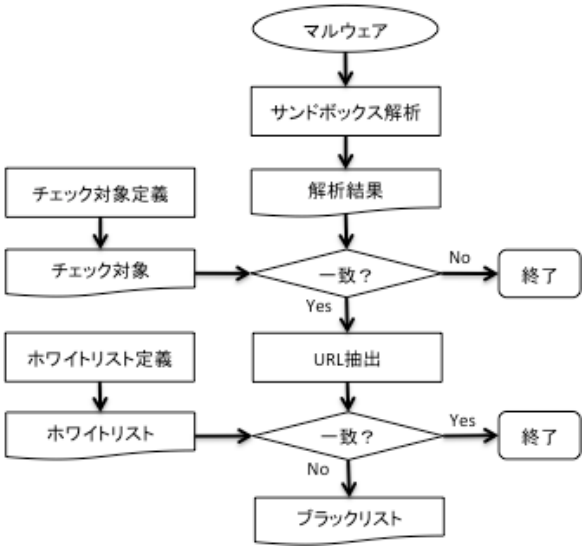

図 1 処理概要

3.2 チェック対象定義

感染ホストの識別情報や，感染ホストを使用しているユーザやアプリケーションの情報を，C&C サーバや他の外部サーバに送信するといったマルウェアの挙動がある. 読み出す情報には以下のようなものがある.

- OS のライセンス情報（シリアルナンバー）
- ユーザ名
- ブラウザの Cookie
- ブラウザの閲覧履歴
- メールサーバ設定(FQDN, IP アドレス，ID/パスワード)
- FTP サーバ設定（FQDN, IP アドレス，ID/パスワード）

これらの情報は通常，レジストリやファイルとして記録されており，チェック対象として記録しておく. 図 2 に具体的なチェック対象のレジストリとファイルの例を挙げる.

```
<Regs>
 <item>HKLM\SOFTWARE\MICROSOFT\WINDOWS NT
\CURRENTVERSION</item>
 <item>HKCU\Software\FTP Explorer\Profiles</item>
  :
</Regs>
<Files>
 <item>C:\Users\user\AppData\Roaming\Mozilla\Firefox\Profiles\</
item>
 <item>C:\Users\user\AppData\Roaming\Microsoft\Windows\Cookies
\Low\user@yahoo[2].txt</item>
  :
</Files>
```

図 2 チェック対象の例

3.3 サンドボックス解析

マルウェアをサンドボックスで解析すると，その解析結果として JSON 形式や XML 形式等で出力される. MWS Datasets 2013 の FFRI Dataset 2013[13]は，Cuckoo Sandbox を用いたマルウェア 2,644 検体の動的解析ログであり，解析対象 1 検体につき 1 ログファイル（JSON 形式）として提供されている.

図 3 に id=560 のログファイルの抜粋(一部 x でマスク済，以降の URL 等記載時にも同様の処理を行う)を示しているが，解析対象ファイルや解析環境のメタ情報，API コール，挙動のサマリとしてのアクセスしたレジストリやファイル，ネットワークアクティビティ等が出力される.

3.4 URL 抽出

FFRI Dataset 2013 のサンドボックス解析結果から，network の http の uri を抽出する. サンドボックス解析結果とチェック対象が一致（あるいは一致した件数が N 件以上）等の条件で，チェック対象の読み出し時刻以降のアクセスであることを条件とする. ただし，uri には図 4 に示すような，URL のスキームとホスト名が重複しているもの（id=1013）や，DGA（Domain Generation Algorithm）[14]のようなホスト名だが，ホスト名に対する DNS クエリのログ（network の dns）もない場合もある（id=1027）.

```
{
  "info": {
    "category": "file",
    "started": "2013-04-09 04:44:53",
    "ended": "2013-04-09 04:47:15",
    "version": "0.5",
 :〈省略〉
"category": "filesystem",
                "status": "FAILURE",
                "return": "0xc0000034",
                "timestamp": "2013-03-28 11:05:10,828",
                "thread_id": "1396",
                "repeated": 0,
                "api": "NtOpenFile",
                "arguments": [
                  {
                    "name": "FileHandle",
                    "value": "0x00000000"
                  },
                  {
                    "name": "DesiredAccess",
                    "value": "0x001200a9"
 :〈省略〉
"summary": {
       "files": [
         "C:\\WINDOWS\\system32\\msctfime.ime",
         "C:\\DOCUME~1\\cuckoo1\\LOCALS~1\\Temp\
\2632645C17E1985396F0033D15EE253F.bin.2.Manifest",
         "C:\\DOCUME~1\\cuckoo1\\LOCALS~1\\Temp\
\2632645C17E1985396F0033D15EE253F.bin.3.Manifest",
         "C:\\DOCUME~1\\cuckoo1\\LOCALS~1\\Temp\
\2632645C17E1985396F0033D15EE253F.bin.Manifest",
         "C:\\DOCUME~1\\cuckoo1\\LOCALS~1\\Temp\
\2632645C17E1985396F0033D15EE253F.bin.Config",
  :
"http": [
       {
         "body": "",
         "uri": "http://imp.xxxxxxxxxxxxxxxx.com/impression.do/?
user_id=5bf96358-336b-4339-
b511-47279b470712&event=setup_run&spsource=engageBDR_downl
oadmanager-US-
direct&subid=(null)&traffic_source=engageBDR&offer_id=downloadm
anager",
         "user-agent": "||Mozilla/5.0 (Windows NT 5.1) AppleWebKit/
537.11 (KHTML, like Gecko) Chrome/23.0.1271.97 Safari/537.11",
         "method": "GET",
         "host": "imp. xxxxxxxxxxxxxxxx.com",
         "version": "1.1",
         "path": "/impression.do/?user_id=5bf96358-336b-4339-
b511-47279b470712&event=setup_run&spsource=engageBDR_downl
oadmanager-US-
direct&subid=(null)&traffic_source=engageBDR&offer_id=downloadm
anager",
         "data": "GET /impression.do/?user_id=5bf96358-336b-4339-
b511-47279b470712&event=setup_run&spsource=engageBDR_downl
oadmanager-US-
direct&subid=(null)&traffic_source=engageBDR&offer_id=do:
 :〈省略〉
```


図 3 サンドボックス解析結果例

3.5 ホワイトリスト定義

インターネット接続確認のためのメジャーな Web サイト(http://www.google.com)や，グローバル IP アドレス確認のための Web サイト(http://checkip.dyndns.org/)をホワイトリストとして予め定義しておく.


```
# id=1013より抜粋
"http": [
       {
         "body": "",
         "uri": "http://app2.xxxxxxxx.com/http://app2. xxxxxxxx.com/
app.asp?prj=5&pid=wsk1&logdata=MacTryCnt:
0&code=&ver=1.0.0.100&appcheck=1",
         "method": "GET",
         "host": "app2.winsoft3.com",
         "version": "1.1",
         "path": "http://app2. xxxxxxxx.com/app.asp?
prj=5&pid=wsk1&logdata=MacTryCnt:
0&code=&ver=1.0.0.100&appcheck=1",
         "data": "GET http://app2. xxxxxxxx.com/app.asp?
prj=5&pid=wsk1&logdata=MacTryCnt:
0&code=&ver=1.0.0.100&appcheck=1 HTTP/1.1\r\nHost: app2.
xxxxxxxx.com\r\n\r\n",
         "port": 80
       },

# id=1027より抜粋
 "http": [
       {
         "body": "",
         "uri": "http://xxxxxx.1ywsk79gm7g3iq9w.com/?cEI17q20=
%96%CB%D2%D3%D6%D5%8F%94%AE%A9%D9%9Fgi%9D
%D3%98%A0f%CF%AA%9A%DD%94%98%A7%A3%93u%82%94%9D
%D3X%A7%E8%A2%E7%E5%CA%C4T%A6%E2%DB%B0%A0nj%9D
%93%A3%D5%9A%A6%DB%9A%A4x%B3%95%DA%CC%A9%86hl
%AAid%AB%AB%A3%A8%AB%B7_u%B2%A8%A6%A0xj%AB%A3y%9Bk
%AD%A6k%BA%60%9A%B5%9D%89%B7%AE%A4t%A2b
%60%A2%95%A2%9F%A2%A4%5Ea%AA%A7%A3%9Bgi%97%9Di%8D
%A9%A6%A1i%A7%5E%93%A0%9F%9Bi",
         "user-agent": "Mozilla/4.0 (compatible; MSIE 8.0; Trident/
4.0; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR
3.0.04506.590; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; .NET CLR
3.0.4506.2152; .NET CLR 3.5.30729)",
         "method": "GET",
         "host": "xxxxxx.1ywsk79gm7g3iq9w.com",
```


図 4 URL 抽出時の注意点の例

3.6 ブラックリスト

サンドボックス解析結果がチェック対象の情報と一致するものがあった場合に，当該マルウェアの通信先 URL から，ホワイトリストに一致した URL を除外してブラックリストとする.

4 データセットを用いた事例調査

提案方式の評価に向けて，FFRI Dataset 2013 を対象に，基礎データの集計結果と事例を挙げ考察する.

4.1 集計

FFRI Dataset 2013 のサンドボックス解析結果のうち，4 件(id=1199, 1496, 1578, 2567)については JSON 形式のファイルが途

中で切れているため集計からは除外し，2,640検体についての各集計値を表 1 に示す．

表 1 基礎データ

項目	件数
HTTP 通信ありの検体	256
延べ URL	1,628
ユニーク URL	628
ユニーク FQDN	147

本データセットにおいては，約 9%の検体のみで HTTP 通信を行っている．延べ URL 数に対するユニーク URL 数は約 38%となっており，同一の URL に複数回アクセスしている検体や，同一あるいは URL クエリパラメータが異なる URL へアクセスしている複数の検体がある．

1 検体あたり 90 秒間の解析時間では不十分（短い）であったり，解析環境を検知して動作を停止する検体があったり，検体を取得してから動的解析までの時間経過によってアクセス可能な通信先が減少したりするため，これらを解決することにより抽出可能な URL 数を増やすことができる可能性がある．

4.2 URL ブラックリスト生成例 1

ブラウザの Cookie や履歴の情報をファイル読み出しのチェック対象とした場合，id=1565 などが URL 抽出対象となり，ブラックリストが生成される（図 5）．パス部が{3 桁の数字}/{3 桁の数字}.html というパターンの URL があることがわかる．当該検体以外でもいくつかの検体で同様のパターンが確認できた．

```
＜チェック対象＞
C:\\Documents and Settings\\cuckoo1\\Cookies\\index.dat
C:\\Documents and Settings\\cuckoo1\\Local Settings\\History\
\History.IE5\\index.dat

＜ブラックリスト＞
http://xxxxxxx.net/826/759.html
http://sp3.xxxxxx.com/?dm=elacyts.net&acc=3F254F8E-
C939-4DF2-84B2-CA2A97E466E5
http://xxxxxxx.net/501/751.html
http://xxxxxxxxxx.com/497/873.html
```

図 5 チェック対象がファイルの場合の例

4.3 URL ブラックリスト生成例 2

インストール済ソフトウェアの情報をレジストリ読み出しのチェック対象とした場合，id=1690 が URL 抽出対象となった（図 6）．ホスト名が IP アドレスの URL も当該検体以外で見られ，中には 80 番ポートを使用せず，他のポートを指定した URL アクセスを行っている検体もある．なお，プロダクトキーの読み出しは本データセットでは見られなかった．

```
＜チェック対象＞
HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\
\CurrentVersion\\Uninstall

＜ブラックリスト＞
http://x.xxx.175.164/content/offers/default.aspx
```

図 6 チェック対象がレジストリの場合の例

5 課題

ここまでで，提案方式とデータセットを用いた事例調査結果を述べたが，本提案方式に関わる主な課題を以下に示す．

（ア） チェック対象のメンテナンス

OS のバージョンやインストールされているアプリケーションの種類によって，チェック対象とすべきファイルやレジストリが変わってくる．また，プロダクトキーや各種設定情報など何をチェック対象とすべきかも，その利用価値や実際のマルウェアの挙動を分析しながらチェック対象をメンテナンスしていく必要がある．

（イ） ホワイトリストのメンテナンス

正常通信に紛れてマルウェアが通信を行うことを考えると，メジャーサイトであってもマルウェアが不正利用を行う場合がある．また，マルウェアがインターネット接続確認を行う場合，メジャーサイトであるとは限らないため，チェック対象のメンテナンスと同様に実際のマルウェアの挙動を分析しながらメンテナンスしていく必要がある．

（ウ） サンドボックス解析結果の詳細度

本研究で用いたデータセットには，アクセスしたファイルやレジストリの一覧だけではなく，API コールの時刻やプロセスツリーなど詳細なデータが利用できるが，その詳細度はサンドボックスに依存する．また，サンドボックスの

インターネットへの接続条件や，マルウェアが備える解析環境の回避技術に対してシステム固有の情報をランダム化する等の実現レベルによって，解析により取得可能な情報量が大きく変わってくることが考えられる．

(エ) ブラックリストとしての確からしさ

今回の提案方式で生成される URL ブラックリストが，テイント解析等による従来手法と比較してどの程度悪性サイトを判定できているのかを確認する必要がある．また，ドメインや IP アドレスのレピュテーション情報などとの比較も一つの方法として考えられる．

(オ) 攻撃耐性

提案方式で URL ブラックリストを生成し検知・防御に活用する場合に、回避策として，サンドボックスで解析するマルウェア検体の役割分担により，ある検体で情報収集を行って感染ホスト上にファイルとして記録し，他の検体がそのファイルを読み出して外部へ送信する等が考えられる．

6 まとめ

本稿では，マルウェアのサンドボックス解析結果をもとに，システム情報やユーザ情報に関するファイルやレジストリの読み取りを条件とした，簡易な URL ブラックリスト生成方式を提案した．そして FFRI Dataset 2013 を用いた事例調査の結果を示すとともに，課題の考察を行った．今後は課題(ア)に対して，精査したチェック対象をもとに，テイント解析結果との比較など有効性の評価を行っていく．

参考文献

[1] AV Comparatives: Whole Product Dynamic “Real-World” Protection Test – (March-June 2013), http://www.av-comparatives.org/wp-content/uploads/2013/07/avc_prot_2013a_en.pdf (参照 2013/08/19)

[2] IPA:「新しいタイプの攻撃」の対策に向けた設計・運用ガイド改訂第 2 版，http://www.ipa.go.jp/security/vuln/newattack.html (参照 2013/08/19)

[3] Palo Alto Networks: URL Filtering, https://www.paloaltonetworks.com/products/features/url-filtering.html (参照 2013/08/19)

[4] DigitalArts : i-Filter , http://www.daj.jp/bs/i-filter/ (参照 2013/08/19)

[5] Akiyama, M., et al.: Design and Implementation of High Interaction Client Honeypot for Drive-by-download Attacks, IEICE Transactions on Communication, Vol.E93-B No.5 pp.1131-1139, May 2010.

[6] 川元, 他:マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価, MWS2011(2011 年 10 月)

[7] 市野, 他:トラヒックの時系列データを考慮した AdaBoost に基づくマルウェア感染検知手法, 情報処理学会論文誌 53(9) 2012 年

[8] 大月, 他:マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価, MWS2012 (2012 年 10 月)

[9] Jacob, G., et al.: Jackstraws: Picking Command and Control Connections from Bot Traffic, 20th Usenix Security Symposium. USA, August 2011.

[10] Kang, G. M., et al.: DTA++: Dynamic Taint Analysis with Targeted Control-Flow Propagation, Proceedings of the 18th Annual Network and Distributed System Security Symposium, Feb. 2011

[11] 川古谷, 他:テイント伝搬に基づく解析対象コードの追跡方法, MWS2012 (2012 年 10 月)

[12] Egele, M., et al.: A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv. 44, 2, Mar. 2008

[13] 神薗, 他:マルウェア対策のための研究用データセット ～MWS Datasets 2013～, MWS2013 (2013 年 10 月)

[14] TrendLabs SECURITY BLOG, マルウェア解析の現場から -06 「DGA」, http://blog.trendmicro.co.jp/archives/3799 (参照 2013/08/19)