G-10

深層学習による画像分類の WindowsPC を対象としたログ解析への応用

山崎 暁† 廣友 雅徳‡ 白石 善明* 福田 洋治†
Akira Yamazaki Masanori Hirotomo Yoshiaki Shiraishi Youji Fukuta

1. はじめに

企業のコンピュータに関する情報管理やシステム運用に関するセキュリティ上の脅威となる出来事や案件のことをセキュリティインシデントと言い，マルウェア感染や不正アクセス，機密情報の流出などがこれにあたる．企業活動における IT の活用は，企業の成長や事業の発展，グローバル化に対応した経営変革のために必要であるとされているが，その一方で，コンピュータへのウイルス感染等，セキュリティインシデントへの対策が重要な問題となっており，情報セキュリティ対策専門のチームである CSIRT(Computer Security Incident Response Team)を組織内に設置する例が増加している[1]．

かつてのセキュリティ対策は，インシデントを発生させないための対策に主眼が置かれていたが，近年のサイバー攻撃は年々高度化・巧妙化の一途を辿っており，被害を確実に防ぐことは不可能である．そのためインシデント発生を防ぐ手立てとともに，インシデントが発生してしまったことを想定し，インシデント後の対応策をあらかじめ準備しておかなければならない．そこで重要になるのが，インシデントが発生した後，システムの被害状況や範囲を明らかにして，速やかにインシデントの原因を分析することである．

セキュリティインシデントの原因を調査する手段の一つとしてログ解析がある．ログ解析を行うことで，誰がどんな操作を行ったのか，システムがどういう状態にあるのか，システムがどんな通信を行ったのか等を時系列に把握することができる．本研究では，Windows PC のイベントログを抽出，これを画像化し，画像分類の深層学習を利用し，攻撃の有無，攻撃が確認された場合，使用されたツールを推測する Windows PC のログ解析の手法を提案する．また，いくつかの攻撃ツールを異なる条件で動作させイベントログを取得，画像化とラベル付けを行い学習，評価のためのデータセットを作成して，深層学習による画像分類の精度を確認する．

2. 関連研究

マルウェア感染などの悪意ある活動の発見には「パターンマッチ(ブラックリスト，ホワイトリスト含む)」，「振る舞い分析」，「イベント相関分析」などの手法を用いるのが一般的であるが，洗練された攻撃手法や未知の攻撃はこれらの手法で解決できない場合がある．既存手法に依存せず，かつ攻撃者が変更しづらい攻撃の本質となる部分を検出の条件にすることができれば，既存手法と組み合わせることで，よりセキュリティレベルを上げることが可能になるという考えから，鈴木ら[2]は深層学習を利用したログ解析の手法を提案している．

†近畿大学，Kindai University
‡佐賀大学，Saga University
*神戸大学，Kobe University

鈴木らの手法では，一般的なファイアウォールや Web プロキシサーバの膨大なログから 1 時間当たり 1 分ごとの通信回数をカウントし，それを 60 ドットの画像に見立てて深層学習を使った画像認識を行うことにより悪性通信を検出する．

他にも機械学習を利用したログ解析の手法が研究されている．小倉ら[3]は DNS クエリログが前後に問い合わせられるドメインによって特徴付けられることに着目し，ネットワーク内の端末が生成する DNS クエリログに Word2Vec を適用することで，クエリログに含まれる全ドメインの特徴ベクトルを導出し，ブラックリストに含まれる既知悪性ドメインの特徴ベクトルとのコサイン類似度が閾値を超えたドメインを未知悪性ドメインとして検出する手法を提案している．

藤野ら[4]は，約 2600 個のマルウェア検体に対してマルウェア動的解析システム Cuckoo Sandbox を適用し，収集した大量の API コールログから Win32 API の関数名と引数の情報を使って Bag-of-Words(BoW)モデルにより特徴ベクトルを作成し，kmeans 法および非負値行列因子分析(NMF)を用いたクラスタ分析を行いマルウェアの分類と特徴抽出を行う手法を提案している．

3. 深層学習による画像分類の WindowsPC を対象としたログ解析への応用

Windows PC を対象とした攻撃の有無と使用されたツールを推測するログ解析の手法の手順を図 1 に示す．以降の節で提案手法の詳細な手順を述べる．

3.1 攻撃ツールを使用した痕跡の抽出

準備として，ログの取得対象の Windows PC に対して，攻撃ツールを実行し，イベントログに残す．Windows イベントログとは，Windows に搭載されている機能であり，Windows 上の様々なユーザ操作，アプリケーションの実行，各種エラー，セキュリティに関する情報といった，Windows 上で発生した様々なイベントが記録されている．

なお，Windows にはもう一つ，イベントトレースログ(Event Trace Log，ETL)と呼ばれるものがある．イベントログが作業をした後の記録であるのに対し，ETL は起動時のプログラムの経過を記録する場合に使われることが多く情報量も多い．しかし，一般的には「ログインした」や「インストールした」といったイベントの記録で十分なことが多く，デバイスドライバの挙動や Windows Update のア

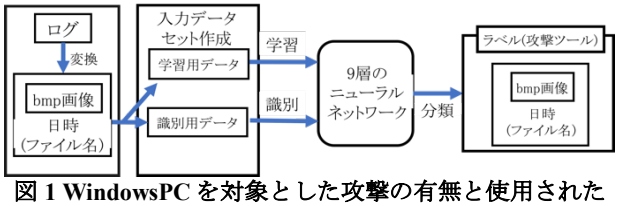

図 1 WindowsPC を対象とした攻撃の有無と使用されたツールを推測するログ解析の手法

ップデート作業の経過などを記録する ETL は日常的な記録には向いていない.

Windows イベントログは.evtx というファイル形式でログを記録しているが，このファイルはバイナリ形式のため，テキストエディタなどでログの内容を確認することができない．イベントログを扱う方法は，

- イベントビューア(GUI)
- wevtutil.exe(コマンドライン)
- WMIC.exe(コマンドライン)
- PowerShell コマンドレット
- Win32API や.NET Framework によるソフトウェア開発

などがあるが，特に一般的な方法は，Windows に用意されている「イベントビューア」というアプリケーションを使う方法である．「イベントビューア」は GUI アプリケーションであり，項目を指定しての並び替え，条件を指定してのフィルター機能などがあり，簡単にログを確認するすることができる．

Windows イベントログには，次のような 3 つのイベント種別があり，種別ごとにファイルが分けられている．

- アプリケーション：
  主にサービスの実行結果やエラーなどを記録する
  [保存場所]
  …%SystemRoot%¥System32¥winevt¥Logs¥Application.evtx
- セキュリティ：
  特定ファイルの読み取りやログオンに関する情報を記録する
  [保存場所]
  …%SystemRoot%¥System32¥winevt¥Logs¥Security.evtx
- システム：
  Windows システムの標準サービス，ディスクドライバ，OS などの記録を格納する
  [保存場所]
  …%SystemRoot%¥System32¥winevt¥Logs¥System.evtx

Windows のイベントログは，ログなどの改ざんを防ぐために，外部ファイルからのアクセスが制限されているが，今回は解析にだけ使用するという名目のもと，権限の変更を行い，外部からのアクセスを可能にする．

JPCERT/CC のインシデント調査のための攻撃ツール等の実行痕跡調査に関する報告書[5]では，実際の攻撃に使われるツールの実行時にどのようなログが残るのか，またどのような設定をすれば十分な情報を含むログを取得できるようになるのかがまとめられている．

本研究では，報告書[5]の中の，ネットワーク内部に侵入した攻撃者が悪用する可能性が高い攻撃ツールが使用された場合に残るイベントログの一覧(「攻撃ツール分析シート[6]」)に記載されているツールのうち，Windows イベントビューアの Sysmon[7]，セキュリティログ，システムログ」に痕跡が確認できるツールを実際に使用した．

Sysmon はマイクロソフト社が提供しているツールであり，端末上で動作したアプリケーションの情報やレジストリエントリの作成，通信など Windows OS の様々な動作をイベントログに記録するツールである．

イベントログから必要な情報を抜き出すプログラムは，イベントログを対象として，そのログファイルを xml 形式で取得し，変数にイベント ID を格納する．格納したイベント ID が攻撃ツール分析シートの実行時に記録される主要な情報として記載されているそれぞれの攻撃ツールで確認できるイベント ID である場合，ノード表に従って，作成した配列の中の数を「1」にする(元々は「0」)．なお，配列は Sysmon，せキュリティログ，システムログそれぞれで別々に作成する．

3.2 ログの画像化

ログを bmp 画像へ変換するために，まず図 2 のようなノード表を作成する．攻撃ツール分析シートには，ツールの実行時に残るイベント ID の順番が記載されている．その情報を参考に，残るとされる痕跡にノードを割り当て，ノード表を作成する．

その後，ノード表のイベント ID と合致するイベント ID をログから探す．合致するイベント ID が見つかった場合，そのイベント ID に応じて bmp 画像に点をプロットし，画像を作成する．例えば，図 3 のようなログがあったとする．

この場合，図 2 のノード表にあったイベント ID と合致するものが 4 つあり，残るログに順番があるという特性を踏まえると，図 4 のように有向グラフを作成することができ，プロットの際は，有向グラフをもとに，bmp 画像の縦軸をノード元，横軸をノード先と定義し，例えばノード 0 からノード 1 へ繋がる場合，画像の(1,0)を白から黒にする．

ログを bmp 画像に変換するプログラムでは，先に作成した配列の情報が出力されているテキストファイルを参照し，その配列から bmp 画像を作成する．はじめに，224*224 の全ての画素が白(252,252,252)の bmp 画像を作成する．その画像に対して，縦軸をノード元，横軸をノード先として，セキュリティログの配列の 0 番目とシステムログの配列の 1 番目を参照し，ともに「1」であれば，bmp 画像の(1,0)を白から黒(0,0,0)にする．その後，その攻撃ツールが使われた日時を取得する．

ログが残る順番

ノード番号	ログ	イベントID
0	セキュリティ	5145
1	システム	7045
2	Sysmon	1
3	Sysmon	1
4	セキュリティ	4689
5	システム	7036
6	Sysmon	11
7	セキュリティ	4674

図 2 ノード表の例

```
2019/3/1 11:11:00 5145
2019/3/1 11:11:23 5
2019/3/1 11:11:27 7045
2019/3/1 15:34:34 1
2019/3/1 15:45:51 13
2019/3/1 15:50:22 1
…
```

図 3 ログの例

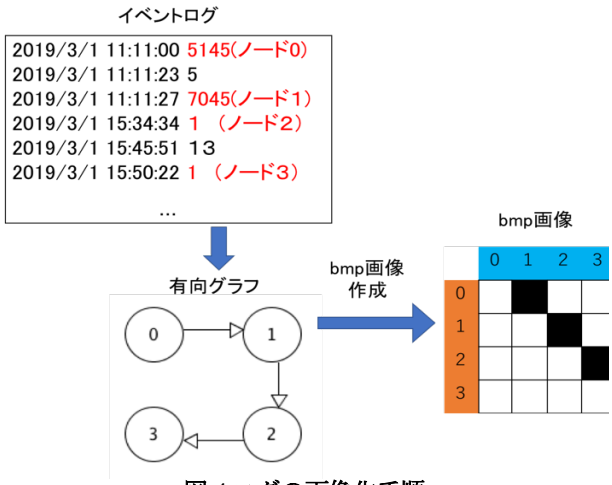

図 4 ログの画像化手順

日時は配列とともに格納されており，全てのログの出力後，その日時を取得し，ファイル名にする．最終的に分類された bmp 画像を見ることで，そのファイル名から，いつ，どんな攻撃ツールが使用されたかがわかる．それにより，インシデントに素早く対応できると考えられる．

3.3 画像分類の深層学習

一般的な 9 層のニューラルネットワークを作成し，深層学習を行う．深層学習には Keras[8]という，Python で書かれたオープンソースネットワークライブラリを用いる．Keras は 2015 年に開発されたライブラリであり，TensorFlow[9]や CNTK[10]，Theano[11]といった他の深層学習ライブラリの上部で動作することができる．迅速な実験を可能にするよう設計されており，使いやすさに重点をおいて開発されている．

畳み込み層には 3*3 のフィルターを使用し，入力と出力の大きさを同じにしている．プーリング層のプーリング領域サイズは 2*2，ストライドを 2*2 としている．プーリング層は MaxPooling を用いる．活性化関数は Leaky ReLU を使用し，出力層には softmax を活性化関数として使用する．Dropout は全結合層の間にセットし，50%の割合でドロップする．

図 5 のように，ラベルなしのデータから推測された出力データと，ラベル付きデータとの誤差を用いて，重みを更新することで学習を行う．また，ネットワークのモデル構造を図 6，各層の出力形式を図 7 に示す．

4. 評価実験

提案手法のプログラムを python で試作し，攻撃者が悪用する可能性の高い攻撃ツール 49 個のうち 5 個，PsExec[12]，wmi[13]，schtasks[14]，wmiexec.vbs[15]，WinRM[16]を実行した時のイベントログから作成した bmp 画像のデータセットに，攻撃ツール未使用時(NoTool)のイベントログから作成した bmp 画像のデータセットを加えた 6 つのデータセットについて学習・推測を行った．

以下の表 1 に，今回使用した 5 つのツールの説明を記載する．1 つの攻撃ツールにつき学習データセットとして bmp 画像 300 枚，バリデーションデータセットとして bmp 画像 300 枚を用意した．

データセットの画像は次の 2 種類がある．

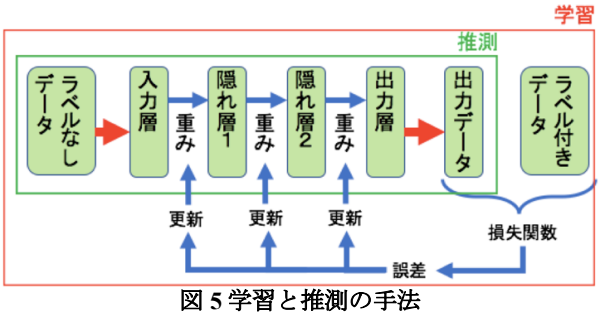

図 5 学習と推測の手法

層番号	入力層	
1	3×3 conv, 32	フィルターの大きさ
2	3×3 conv, 32	フィルター数
	maxpool, /2	
3	3×3 conv, 64	
4	3×3 conv, 64	
	maxpool, /2	ストライドの大きさ
5	3×3 conv, 64	
6	3×3 conv, 64	
	maxpool, /2	
7	fc 1024	ユニット数
8	fc 1024	
9	fc6	(softmax)

図 6 モデル構造

層番号	枚数	大きさ	
1	32	224×224	
2	32	224×224	
	32	112×112	
3	64	112×112	
4	64	112×112	
	64	56×56	
5	128	56×56	
6	128	56×56	
	128	28×28	1次元に変換
7	–	1024	
8	–	1024	
9	–	6	

図 7 各層の出力形式

A) 実際に攻撃ツールを使用し，動的解析によりログから作成した画像

B) 攻撃ツールが使用された場合に残る痕跡からプロットされるはずの画素を白から黒にし，他の画素に対して乱数でプロットした画像

表 1 使用した攻撃ツール

ツール	ツール概要
PsExec	説明： リモートホスト上でコマンドを実行する. 攻撃時における想定利用例： ドメイン内のホストやサーバでリモートコマンドを実行する.
wmi	説明： Windows のシステム管理に使用する. 攻撃時における想定利用例： WMI を用いてリモートシステムの情報取得や，コマンド実行を行う.
schtasks	説明： 指定した時刻にタスクを実行する. 攻撃時における想定利用例： ユーザに気づかれないように実行ファイルやスクリプトを，任意のタイミングで実行する.
wmiexec.vbs	説明： Windows のシステム管理に使用する. 攻撃時における想定利用例： リモートホストで，スクリプトを実行する.
WinRM	説明： 遠隔端末から情報を搾取する. 攻撃時における想定利用例： リモートコマンドを実行する前に調査のため実施する.

A)の画像を作成する際は，以下のような手順で行う.

1) 一定時間(10 分)おきに攻撃用 PC から標的用 PC に対して攻撃ツールを実行する．組織でのインシデント対応を想定しているため，標的用 PC ではファイル操作などの通常操作を行うこととする．これにより，標的用 PC のログには攻撃ツールの痕跡と通常操作の痕跡が残る.
2) ログに対して，ログを画像化するプログラムを実行し，画像を作成する.
3) 画像化後，異なるデータを作成するため，標的用 PC のログを削除する.

上記の処理を繰り返し，使用したツールのラベル付きデータを作成する.

先に述べた手順を行って，A)の画像を 30 枚，B)の画像を 30 枚の合計 60 枚をそれぞれの攻撃ツールで用意した．これらの画像を 5 倍に拡張(Data Augmentation)し，1 つの攻撃ツールに対してランダムに振り分けることで，学習データセットとバリデーションデータセットが各々300 枚になるようにした.

また，テストデータセットとして A)の画像を 480 枚用意し，その画像を 5 倍に拡張した.

評価については，ログ(bmp 画像)の分類精度で行う．ログの解析対象の Windows PC の動作環境を表 2，深層学習を行う評価環境を表 3 に記載する.

本研究で作成したノード表は表 4 の通りである．また，有向グラフは図 5 のようになった.

表 2 解析環境

CPU	Intel Core i3-7100U
GPU	----
メモリ	16GB
OS	Windows 10 Pro

表 3 評価環境

CPU	Intel Core i5
GPU	GeForce GTX 1070
メモリ	8192MB
OS	Ubuntu 16.04 LTS

表 4 本研究で作成したノード表

ツール名	ノード番号	ログ	イベント ID
PsExec	0	セキュリティ	5145
	1	システム	7045
	2	Sysmon	1
	3	Sysmon	1
	4	セキュリティ	4689
	5	システム	7036
	6	Sysmon	11
	7	セキュリティ	4674
wmi	8	Sysmon	3
	9	Sysmon	1
schtasks	10	Sysmon	3
	11	Sysmon	1
	12	Sysmon	1
	13	Sysmon	13
wmiexec.vbs	14	Sysmon	3
	15	セキュリティ	4672
	16	セキュリティ	5142
	17	Sysmon	1
	18	セキュリティ	4663
	19	セキュリティ	4656
	20	セキュリティ	5144
WinRM	24	セキュリティ	4624
	25	セキュリティ	4634

バリデーションデータセットの推測値と，バリデーションデータセットの教師データから正解率を計算し，一番高かったエポック数のモデルの重みを利用し，推測を行った．各攻撃ツールのテスト画像 480 枚に対し 10 回（データ拡張倍数×ホールドアウト検証回数(2 回)）推測し，正解数と不正解数を算出して正解率を計算した．この結果を表 6 に示す.

評価実験の結果，wmi と，何も攻撃ツールを使っていない NoTool を除いた PsExec，schtasks，wmiexec.vbs，WinRM の 4 つの攻撃ツールについては 90%を超える非常に高い正解率が得られた.

しかし，それと対照に wmi については正解率が 68%となり，7 割近くという結果ではあるが，他の攻撃ツールと比較すると低い結果となった．このような結果になった原因には，ノード数が関係していると考えられる．他の攻撃ツールはノード数が最低でも 4 つあるのに対し，wmi の有向グラフはノード数が 2 つであり，これを元に作成された bmp 画像

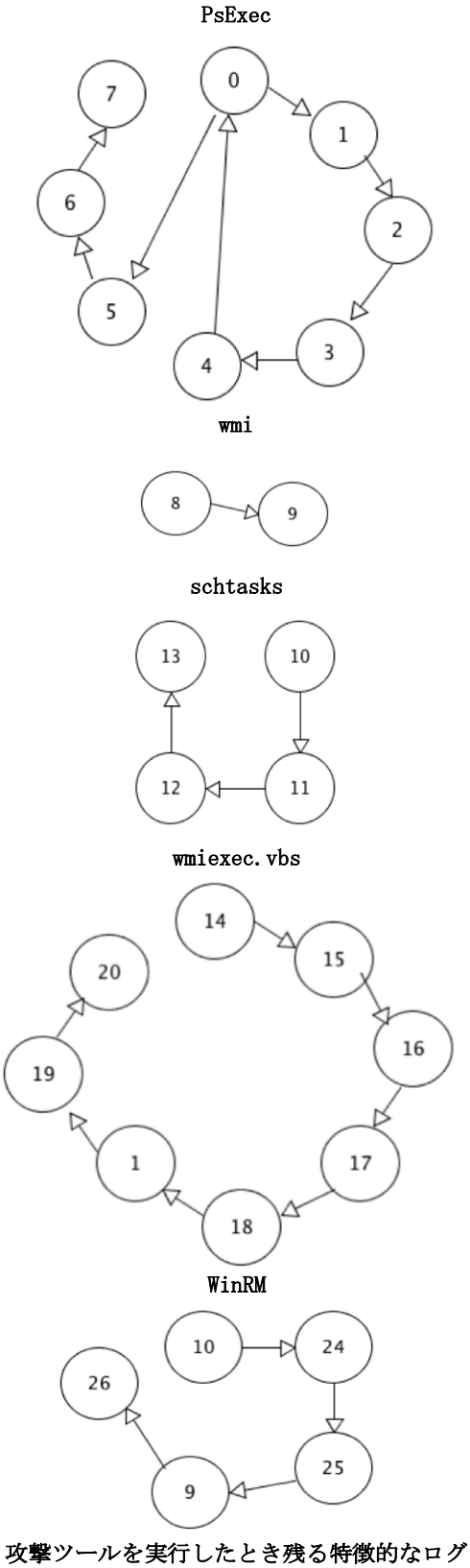

図 5 攻撃ツールを実行したとき残る特徴的なログのノードを時系列につないだ有向グラフ

では一箇所の画素のみが白から黒になるだけであるため，画像としての特徴が弱く，他のノード数が多い攻撃ツールより正解率が低くなったと考えられる．

表 6 イベントログの画像分類の評価実験の結果

使用ツール種別	正解数	不正解数	正解率
PsExec	471	9	98%
wmi	330	150	68%
schtasks	467	13	97%
wmiexec.vbs	477	3	99%
WinRM	480	0	100%
NoTool	373	107	77%

5. おわりに

本研究では，深層学習による画像分類の Windows PC を対象としたログ解析の応用として，Windows イベントログから bmp 画像を作成し，深層学習を利用することで攻撃の有無と使用されたツールを推測するログ解析の手法を提案し，実験により画像分類の精度を確認した．

本研究で提案したログ解析の手法は，JPCERT/CC が公開しているインシデント調査のための攻撃ツール等の実行痕跡調査に関する報告書[5]の中の攻撃ツール分析シート[6]にまとめられている一部の攻撃ツールの実行痕跡を分類するものであり，これにより一部ではあるが，使われた攻撃ツールの判別が可能であることを確認した．

ログの分類を画像で行う事により，攻撃ツールによってどのような被害を受けたのか，その攻撃の段階を分類された画像を人が見て確認する事によってわかる点は，判別を画像で行うことのメリットの 1 つと考えられる．

今後の課題として，本研究で行った実験では全体的に正解率が高かったものの，分類する攻撃ツールを増やした場合，正解率が低くなることが懸念される．

攻撃ツールを実行したとき残る特徴的なログに対応するノード数を増やすことができれば，画像に変換した際に他の画像との差がつきやすくなり，正解率の向上を図ることができると考えられる．

参考文献

[1] IPA：企業の CISO や CSIRT に関する実態調査，<https://www.ipa.go.jp/files/000058850.pdf>，(参照 2020-07-21).

[2] 鈴木博志，梨和久雄：ディープラーニングを用いたログ解析による悪性通信の検出，Internet Infrastructure Review(IIR)，vol.42，pp10-17(2019).

[3] 小倉光貴，佐藤彰洋，中村豊，野林大起，池永全志：Word2Vec を利用した DNS クエリログ解析による未知悪性ドメインの検出，2019 年度電気・情報関係学九州支部連合大会講演論文集，06-2A-03(2019).

[4] 藤野朗稚，森達哉：自動化されたマルウェア動的解析システムで収集した大量 API コールログの分析，コンピュータセキュリティシンポジウム 2013 論文集，vol.4，pp618-625(2013).

[5] JPCERT コーディネーションセンター：インシデント調査のための攻撃ツール等の実行痕跡調査に関する報告書，<https://www.jpcert.or.jp/research/ir_research.html>，(参照 2020-07-21).

[6] JPCERT コーディネーションセンター：ツール分析シート，
<https://jpcertcc.github.io/ToolAnalysisResultSheet_jp>，(参照 2020-07-21).

[7] Microsoft：Sysmon – Windows Sysinternals | Microsoft Docs，<https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>，(参照 2020-07-22).

[8] keras-team：Keras Documentation，
<https://keras.io/ja/>，(参照 2020-07-22).

[9] Google：TensorFlor.org，
<https://www.tensorflow.org/>，(参照 2020-07-22).

[10] Microsoft：Cognitive Toolkit – CNTK - Microsoft Docs，<https://docs.microsoft.com/ja-jp/cognitive-toolkit/>，(参照 2020-07-22).

[11] Theano Development Team：Welcome – Theano 1.0.0 documentation – Deep Learning，
<http://deeplearning.net/software/theano/>，(参照 2020-07-22).

[12] Microsoft：PsExec – Windows Sysinternals | Microsoft Docs，<https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>，(参照 2020-07-22).

[13] Microsoft：WMI Reference – Microsoft Docs，
<https://docs.microsoft.com/ja-jp/windows/win32/wmisdk/wmi-reference>，(参照 2020-07-22).

[14] Microsoft：schtasks – Microsoft Docs，
<https://docs.microsoft.com/ja-jp/windows-server/administration/windows-commands/schtasks>，(参照 2020-07-22).

[15] Twi1ight：wmiexec.vbs - GitHub，
<https://github.com/Twi1ight/AD-Pentest-Script/blob/master/wmiexec.vbs>，(参照 2020-07-22).

[16] Microsoft：Windows Remote Management – Win32 apps | Microsoft Docs，<https://docs.microsoft.com/en-us/windows/win32/winrm/portal>，(参照 2020-07-22).