Computer Security Symposium 2017
23 - 25 October 2017

ランサムウェア検知のための特徴解析

重田 貴成[1]　伊沢 亮一[2]　森井 昌克[1]　井上 大介[2]　中尾 康二[2]

概要：近年，ランサムウェアの流行に伴い，ランサムウェア検知手法の精度向上が要求されている．本稿では，ランサムウェア検知のために必要となる特徴を調査するために，ランサムウェアとワーム，正規ソフトウェアの VirusTotal レポートを解析した結果を考察する．調査では DLL インジェクション，リンクされた DLL，ファイル入出力，実行ファイルの署名に着目する．その結果，ランサムウェアとワーム，正規ソフトウェアの間で特徴的な差異が存在することを確認できた．これらの結果をもとにランサムウェア検知手法を改良することで，ランサムウェアの検知精度が向上すると考えられる．

キーワード：ランサムウェア，VirusTotal，ランサムウェア検知

Analysis of Ransomware Characteristics for Detection

Takanari Shigeta[1]　Ryoichi Isawa[2]　Masakatu Morii[1]　Daisuke Inoue[2]　Koji Nakao[2]

Abstract: In recent years, it is required to improve the accuracy of ransomware detection because of the spread of ransomware. In this paper, to research the features for the detection of ransomware, we consider the VirusTotal report of ransomwares, worms, and benign softwares. We focus on DLL injection, linked DLL, file I/O, and signatures of EXE files. As a result, there are distinctive differences between ransomwares, worms, and benign softwares. We think that the accuracy of ransomware detection can be improved by this result.

Keywords: ransomware, VirusTotal, ransomware detection

1. はじめに

ランサムウェアはコンピュータ内のファイルを，復号鍵を知らせることなく暗号化し，ファイルの所有者が復号できなくなることで被害を及ぼす．近年，ランサムウェアの流行が深刻な問題となっている．情報処理推進機構から発表された情報セキュリティ 10 大脅威 2017 [1] では，2 位にランサムウェアを使った詐欺・恐喝が入っている．そのため，ランサムウェアの早期発見や被害防止のため，UNVEIL [2]，CryptoDrop [3] などのランサムウェア検知手法が多数提案されている．これらの検知手法では，ファイル入出力，暗号化，支払い画面の表示をランサムウェアの特徴として利用している．さらにランサムウェアの特徴を利用することで，検知手法の改良が可能であると考えられる．

そこで，本稿ではランサムウェアの検知手法の精度向上のために，ランサムウェアの特徴を調査する．VirusTotal のレポートを解析して，ランサムウェアとワーム，正規ソフトウェアを比較する．まず，マルウェアがコード隠蔽のために DLL インジェクションすることに着目し，DLL インジェクションの有無とインジェクション先を比較する．次に，ランサムウェアがパッカーや暗号化ライブラリを利用することに着目し，リンクする DLL の数や種類を比較する．そして，ランサムウェアはファイルの暗号化の際に多くの数や種類のファイルにアクセスすることに着目し，ファイル入出力の回数と種類を比較する．さらに，マルウェアは実行ファイル署名のためのコードサイニング証明書を取得することが難しいことに着目し，実行ファイルの

[1] 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
[2] 国立研究開発法人情報通信研究機構
National Institute of Information and Communications Technology

署名の有無と検証結果について比較する．これらの結果をもとにランサムウェア検知手法を改良することで，ランサムウェアの早期発見や被害防止の精度が向上すると考えられる．

2. 基礎知識

本章では検査レポートを取得した VirusTotal とそのレポートに含まれる項目に関する基礎知識を説明する．

2.1 VirusTotal

VirusTotal [4] はオンラインのマルウェア検査サービスである．検査対象のファイルを VirusTotal にアップロードすると，60 種類以上のアンチウイルスソフトによるスキャンが行われ，各アンチウイルスソフトの悪性判定（マルウェアか否か）の結果が得られる．また，アップロードしたファイルに対して静的解析（リンクされているライブラリ名の取得など）や動的解析（読み書きしたファイル名の取得など）が行われ，それらの結果も取得することができる．

VirusTotal の特徴のひとつとして，過去にアップロードされたファイルの検査結果を蓄積していることが挙げられる．VirusTotal が提供している Private API [5] を用いて，ファイルのハッシュ値（SHA256 [6] など）で検索をかけると，そのファイルがいずれかのユーザによりアップロードされたことがあるとその検査結果が取得できる．ただし，ある一定期間内にアップロードされたファイルのみ検索対象となる．ハッシュ値による検索に加え，「ransomware」のようなキーワードによる検索も可能であり，本研究が行なっているようなランサムウェアの傾向・特徴を調査するといった目的でも VirusTotal を利用することができる．検索には Web API（Application Programming Interface）[7] が用意されている．

2.2 暗号化 API・ライブラリ

ランサムウェアはファイルを暗号化する．この暗号化はオペレーティングシステム（OS）が提供する API や暗号化ライブラリを利用することで可能となる．OS の暗号化 API として Windows の Microsoft CryptoAPI [8] が挙げられる．Windows 95 から組込まれている API であり，Windows 95 以降の Windows のバージョンで利用できる．暗号化ライブラリとしては OpenSSL [9] があり，AES であれば aes_encrypt の関数を呼び出すことで AES が使用できる．OpenSSL の他にも Crypto++ [10] などの暗号化ライブラリがある．ランサムウェアが暗号化 API や暗号化ライブラリの利用の有無はライブラリへのリンクやライブラリのロード，ライブラリに含まれる暗号化関数のコールなどから判定可能である．

2.3 DLL

Dynamic Link Library（DLL）とは実行ファイルの実行時にリンクされるライブラリのことである．DLL には関数のコードが格納されており，実行ファイルが DLL をリンクした後，目的の関数を呼び出す．

DLL をリンクする方法として，暗黙的リンクと明示的リンクの 2 種類がある．暗黙的リンクでは，コンパイル時に実行ファイルのヘッダに DLL の情報を書き込む．この情報を OS が実行ファイルの初期化処理中に読み取り，自動で DLL のロード処理を行う．対して，明示的リンクではプログラムコード中に DLL の情報を書き込む．この場合，OS による初期化処理後の任意のタイミングで，プログラムコードから DLL のロード処理を行う．DLL のロードには LoadLibrary 関数を用いる．

実行ファイルがリンクしているライブラリはヘッダもしくは LoadLibrary 関数の引数などから得ることが可能である．

2.4 DLL インジェクション

DLL インジェクション [11] とは，DLL として実行させたいコードを実装し，この DLL を実行ファイルとは別のプロセスにロードさせることで，任意のコードを実行する手法である．この手法を用いると，エクスプローラなどの Microsoft 製ソフトウェア上で任意のコードを実行できる．そのため，マルウェアの中にはアンチウイルスソフトからの検知を回避するために DLL インジェクションを用いるものが存在する．

2.5 コードサイニング証明書

Windows では実行ファイルに署名をつけることで開発元の証明や改ざん検知を行っている．このとき利用される証明書をコードサイニング証明書と呼ぶ．コードサイニング証明書は鎖状構造になっており，ルート証明書から信頼性が連鎖する形で構成されている．そのため，証明書の検証を行う場合は，まず OS やプログラム中にあらかじめ信頼できるルート証明書を組み込んでおく必要がある．そして，親の証明書を用いて子の証明書を検証するという処理を繰り返し，最終的にコードサイニング証明書の信頼性を確認できるようになっている．

3. 特徴解析の方法と結果

本章では，VirusTotal のレポートに含まれる項目別の比較の結果をもとに，ランサムウェアと正規のソフトウェアの差異を明らかにする．また，ランサムウェアと同じくマルウェアの一つであるワームとも比較する．

3.1 データセット

VirusTotal に対して「ransomware」および「worm」で

表 1　使用レポート数
Table 1　the number of samples

種類	レポート数	ユニークレポート数
ランサムウェア	27776	25153
ワーム	4264	3985
正規ソフトウェア	329	329

検索して，それぞれランサムウェア 27776 検体のレポートとワーム 4264 検体のレポートを取得した[*1]．正規ソフトウェアとして Windows 7 のインストール直後からインストールされている calc.exe などの EXE ファイルの検査レポートを VirusTotal から取得した．これにより，ランサムウェア，ワーム，正規ソフトウェアの 3 種類のレポートが取得でき，それぞれの数を**表 1** にまとめる．

レポートに記載されている検体のハッシュ値（SHA256）からいずれの検体にもハッシュ値の重複がないことを確認した．ただし，ハッシュ値ではある検体の 1 ビットでも変更すると，ハッシュ値が異なってしまい，異なる検体と判定されてしまう．そこで，レポートに記載されている SSDEEP（FuzzyHashing の一つ）[12] で重複を排除したところ，ランサムウェアが 25153 個，ワームが 3985 個，正規のソフトウェアが 329 個となった．よって，SSDEEP の重複を排除せずに調査をしても十分汎用性のある結果が得られると考え，本稿の調査では SSDEEP に関係なくすべてのレポートを用いた．

3.2　対応ビット

検体が Win32 EXE か Win64 EXE かについて調査した．ランサムウェアに関しては Win64 EXE は 1 つのみで，それ以外のすべての検体が Win32 EXE であった．また，ワームはすべての検体が Win32 EXE であった．VirusTotal にアップロードされている検体に関してはワームとランサムウェアは Win32 EXE がほぼ全てであり，差異は認められなかった．Win32 EXE は 32 ビット OS と 64 ビット OS の両方で利用できるのに対し，Win64 EXE は 64 ビット OS でしか利用できないため，多くのコンピュータに感染させたいマルウェアには不適当だからだと考えられる．以上より，EXE の対応ビットからワームとランサムウェアの区別はできない．

3.3　DLL インジェクション

マルウェアは，DLL インジェクションをすることで正規のプログラムに偽装し，アンチウイルスソフトの検知から逃れようとする．そこで，検体が DLL インジェクションする場合に，そのインジェクション先プロセスを調査し，ランサムウェアとワーム，正規ソフトウェアに差異があるか確認した．その結果を**表 2** に示す．DLL インジェクションをしない検体の割合をマルウェアと正規ソフトウェアを比較すると，正規ソフトウェアの 95.1 パーセントに対して，ランサムウェアは 41.5 パーセント，ワームは 70.7 パーセントであった．また，ランサムウェアの rundll32.exe に対するインジェクションの割合 33.5 パーセントは，ワームの 1.0 パーセントや正規ソフトウェアの 0.0 パーセントと比較して非常に大きい．rundll32.exe は Windows 付属ソフトウェアの一つであるが，常時起動しているものではないため，ランサムウェア自身が rundll32.exe を起動させて DLL インジェクションを行っている．以上より，DLL インジェクションを行っている場合はマルウェアの可能性が高い．また，インジェクション先が rundll32.exe の場合はランサムウェアの可能性が高い．

インジェクション先プロセスの概要を次に示す．rundll32.exe は DLL 内のコードを実行する EXE，wmiprvse.exe は WMI 関連の EXE，iexplore.exe は Internet Explorer，notepad.exe はメモ帳，winver.exe は Windows のバージョン情報を表示する EXE，dwwin.exe は Windows のエラー検出する EXE，explorer.exe はエクスプローラー，wscntfy.exe は Windows セキュリティセンター，cmd.exe はコマンドプロンプト，wmiadap.exe は WMI 関連の EXE，vboxtray.exe は VirutalBox Guest Addition である．

表 2　DLL インジェクション先
Table 2　target of DLL injection

インジェクション先	割合 [%]		
	ランサムウェア	ワーム	正規
rundll32.exe	33.5	1.0	0.0
wmiprvse.exe	2.7	0.5	0.6
iexplore.exe	2.5	1.5	0.0
notepad.exe	2.0	13.1	0.0
winver.exe	1.8	0.0	0.0
dwwin.exe	1.5	2.1	2.4
explorer.exe	0.9	0.2	0.0
wscntfy.exe	0.4	0.0	0.0
cmd.exe	0.4	0.6	0.0
wmiadap.exe	0.1	0.0	0.0
vboxtray.exe	0.1	0.0	0.0
その他の exe	12.6	10.2	1.8
なし	41.5	70.7	95.1

3.4　リンクされた DLL

ランサムウェアはパッカーや暗号化ライブラリを利用するため，リンクしている DLL の数や種類が他のソフトウェアと異なると考えられる．そこで，検体がリンクしている DLL を調査し，ランサムウェアとワーム，正規ソフトウェ

[*1] 具体的には VirusTotal の Private API search（`https://www.virustotal.com/vtapi/v2/file/search`）のパラメータの “query” に「ransomware」をセットして検索をかけ，Private API report(`https://www.virustotal.com/vtapi/v2/file/report`) のパラメータに取得した検体のハッシュ値をセットして，検体のレポートを取得した．同様に「worm」でも検索をかけた．

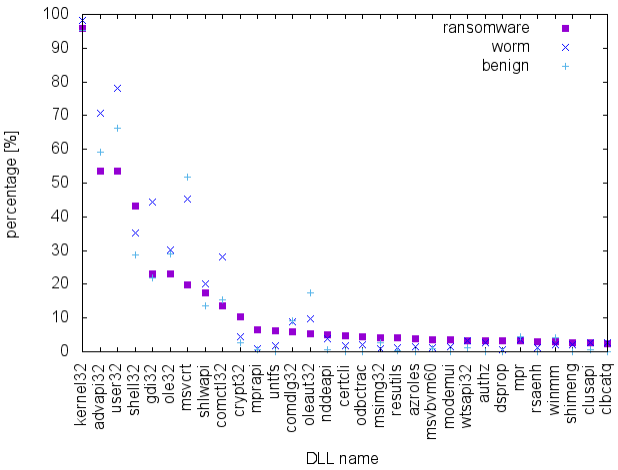

図 1 暗黙的リンクされた DLL
Fig. 1 implicitly linked DLL

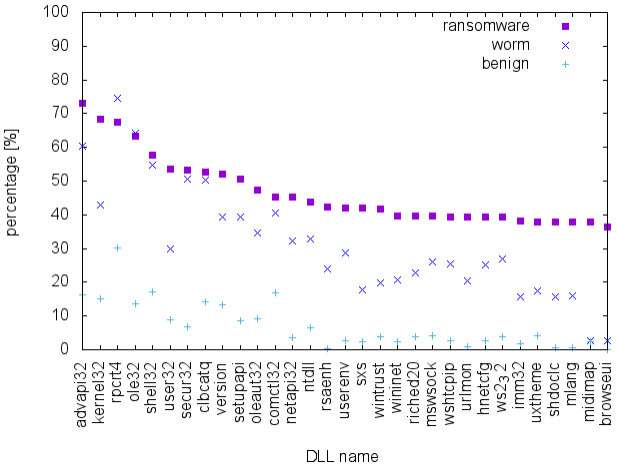

図 2 明示的リンクされた DLL
Fig. 2 explicitly linked DLL

アに差異があるか確認した．検体が暗黙的リンクした DLL を**図 1**，明示的リンクした DLL を**図 2** に示す．暗黙的リンクした DLL に関して，差異は認められなかった．しかし，明示的リンクした DLL に関しては，正規ソフトウェアと比較して，マルウェアは明示的リンクを行う割合が多いことが確認できた．これは，アンチウイルスソフト対策のため，マルウェアがパッカー等で DLL の情報を隠蔽しているためだと考えられる．

一部のマルウェアでは明示的リンクの際に，LoadLibrary 関数に DLL 名の後にスペースを挿入した引数を渡していた．LoadLibrary 関数では DLL 名の後のスペースを無視する仕様となっているため，たとえば “user32 .dll” が引数として渡された場合は user32.dll がリンクされる．本調査のデータセットでは，スペースを挿入した DLL 名で明示的リンクする正規ソフトウェアは存在しなかった．よって，マルウェアがスペースを挿入した DLL 名で明示的リンクするのは，アンチウイルスソフト対策のためだと考えられる．以上より，スペースを挿入した DLL 名で明示的リンクしている場合は，マルウェアの可能性が高い．

表 3 ファイルがオープンされた回数
Table 3 the number of file opening

回数	割合 [%]		
	ランサムウェア	ワーム	正規
0	11.3	11.5	47.4
1	19.0	14.5	11.6
2	3.6	2.7	12.2
3	2.0	2.2	10.6
4	0.8	0.7	2.4
5	2.9	6.7	2.4
6	9.9	12.1	4.3
7	3.3	14.9	2.1
8	2.2	3.6	0.6
9	1.6	3.6	1.5
10 - 99	3.4	7.4	4.3
100 - 999	2.8	1.0	0.6
1000 -	37.1	19.2	0.0

表 4 ファイルがリードされた回数
Table 4 the number of file reading

回数	割合 [%]		
	ランサムウェア	ワーム	正規
0	17.5	16.7	74.2
1	24.7	26.2	19.1
2	15.5	32.8	2.7
3	0.9	0.8	0.9
4	0.7	0.7	1.2
5 - 99	3.4	3.9	1.8
100 - 999	0.6	1.8	0.0
1000 -	36.7	17.1	0.0

多くのランサムウェアが既存の暗号化ライブラリを利用する．そこで，検体が暗号化ライブラリの一つである CryptoAPI を利用しているか否かをレポートから調査した．暗号化ライブラリの一つである CryptoAPI は advapi32.dll 内の API である．しかし，advapi32.dll にはレジストリ操作 API 等も含まれるため，正規ソフトウェアも含めて多くの用途で利用されており，レポートからは CryptoAPI の利用までは判断できなかった．よって，VirusTotal のレポートからは CryptoAPI を利用しているか否かは調査できず，独自で用意した動的解析環境で解析する必要がある．

3.5 ファイル入出力

ランサムウェアは無断でユーザファイルの暗号化を行うため，ワームや正規ソフトウェアと比較してファイルの読み書きが多い．そこで，サンプルのファイル読み書き回数を調べ，ランサムウェアとワーム，正規ソフトウェアで差異があるか確認した．ファイルオープンの回数を**表 3**，ファイルリードの回数を**表 4**，ファイルライトの回数を**表 5** に

表 5　ファイルがライトされた回数
Table 5　the number of file writing

回数	割合 [%]		
	ランサムウェア	ワーム	正規
0	39.9	42.9	91.2
1	7.9	11.9	6.1
2	10.3	21.3	1.5
3	0.9	0.6	0.3
4	0.4	0.2	0.0
5 - 99	3.4	3.7	0.9
100 - 999	0.6	2.1	0.0
1000 -	36.7	17.3	0.0

表 6　リードされた拡張子の種類数
Table 6　the number of read file extensions

種類数	割合 [%]		
	ランサムウェア	ワーム	正規
0 - 9	62.7	79.0	99.7
10 - 19	37.0	20.8	0.3
20 -	0.2	0.2	0.0

表 7　ライトされた拡張子の種類数
Table 7　the number of written file extensions

種類数	割合 [%]		
	ランサムウェア	ワーム	正規
0 - 9	62.8	80.1	99.7
10 - 19	37.0	19.7	0.3
20 -	0.2	0.3	0.0

示す．この結果より，ランサムウェアはワームや正規ソフトウェアと比較して 1000 回以上アクセスされる割合が多いことを確認できた．以上より，ファイルへのアクセス回数が十分多い場合，ランサムウェアであると判断できる．

ランサムウェアは多くの種類のユーザファイルを暗号化するため，他のソフトウェアと比較して読み書きする拡張子の種類は多い．また，ランサムウェアは暗号化により拡張子を同一のものに変更するため，リードする拡張子の種類に対してライトする拡張子の種類は少なくなると考えられる．そこで，リード・ライトされた拡張子の種類を数えた．その結果，リードされた拡張子の種類数は**表 6**，ライトされた拡張子の種類数は**表 7** のようになった．この結果から，正規ソフトウェアと比較してワームやランサムウェアの読み書きする拡張子の数が多いことが確認できた．しかし，ワームとランサムウェアの間で読み書きする拡張子数の差異は認められなかった．また，リードとライトの拡張子数を比較しても，リードする拡張子の種類に対してライトする拡張子の種類は少なくなるということは確認できなかった．これは，VirusTotal の動的解析環境で十分な数のユーザファイルが存在しないためだと考えられる．よって，十分な数のユーザファイルが存在しない環境下では，拡張子の種類数からランサムウェアの判定はできない．

表 8　実行ファイルの署名
Table 8　code signing of exe files

状態	割合 [%]		
	ランサムウェア	ワーム	正規
署名なし	99.5	95.5	76.0
正規の署名	0.4	4.2	21.9
不正な署名	0.1	0.3	2.1

3.6　実行ファイルの署名

一部のアンチウイルスソフトは正規ソフトウェアをマルウェアと誤検知することを防ぐため，署名を用いたホワイトリストを利用している．そこで，サンプルが DLL インジェクションする場合に，そのインジェクション先プロセスを調べた．その結果を**表 8** に示す．実行ファイルに署名されていない検体は，ランサムウェアで 99.5 パーセント，ワームで 95.5 パーセント，正規ソフトウェアで 76.0 パーセントであった．このように，正規ソフトウェアと比較してランサムウェアやワームでは実行ファイルに署名されていない割合が大きいことを確認した．しかし，有効なコードサイニング証明書で署名されているマルウェアも存在し，ランサムウェアは 0.4 パーセント，ワームは 4.2 パーセントであった．以上より，正規ソフトウェアと比較してマルウェアはコードサイニング証明書による署名がされていない割合が多いといえる．しかし，署名のついたマルウェアも存在するため，署名を用いたホワイトリストを利用する際は，信頼できるルート証明書を限定する必要がある．

4.　関連研究

UNVEIL [2] は解析環境でランサムウェアとそれ以外のマルウェアを区別する手法である．UNVEIL はファイル入出力と支払い画面に着目した検知手法である．ランサムウェアはファイルの暗号化を行うため，読み込むファイルのエントロピーに対して書き込むファイルのエントロピーが高い．また，ランサムウェアのファイルアクセスパターンは「上書き」,「リード → 暗号化 → 削除」,「リード → 暗号化 → 上書き」に分類できる．さらに，ランサムウェアの実行後はスクリーンショット中に支払い催促の文面が存在し，実行前のスクリーンショットと比較して類似性が低い．UNVEIL はこれらの特徴を利用してランサムウェアと判定している．

CryptoDrop [3] はランサムウェアを早期検知して，被害を最小限に抑える手法である．CryptoDrop では検知のためにファイル入出力を比較し，ファイルのマジックナンバー，ファイルの類似性，エントロピーの差異によってランサムウェアを判定する．また，判定の際にファイル削除やファイルの種類の差異も考慮している．

5. まとめ

本稿では，ランサムウェア検知のために必要となる特徴を調査するために，ランサムウェアとワーム，正規ソフトウェアの VirusTotal レポートを解析した結果を考察した．まず，マルウェアがコード隠蔽のために DLL インジェクションすることに着目し，DLL インジェクションの有無とインジェクション先を比較した．その結果，DLL インジェクションを行っている場合はマルウェアの可能性が高いこと，インジェクション先が rundll32.exe の場合はランサムウェアの可能性が高いことを確認できた．次に，ランサムウェアがパッカーや暗号化ライブラリを利用することに着目し，リンクする DLL の数や種類を比較した．その結果，正規ソフトウェアと比較してマルウェアは明示的リンクを行う割合が多いこと，スペースを挿入した DLL 名で明示的リンクしている場合はマルウェアの可能性が高いことを確認できた．そして，ランサムウェアはファイルの暗号化の際に多くの数や種類のファイルにアクセスすることに着目し，ファイル入出力の回数と種類を比較した．その結果，ランサムウェアはファイルへのアクセス回数が多いことを確認できた．さらに，マルウェアは実行ファイル署名のためのコードサイニング証明書を取得することが難しいことに着目し，実行ファイルの署名の有無と検証結果について比較した．その結果，正規ソフトウェアと比較してマルウェアはコードサイニング証明書による署名がされていない割合が多いこと，正規の署名をされたマルウェアも存在することを確認できた．これらの結果をもとにランサムウェア検知手法を改良することで，ランサムウェアの早期発見や被害防止の精度が向上すると考えられる．

本研究では VirusTotal の動的解析環境の制限により，暗号化ライブラリの利用や入出力ファイルの拡張子に関して十分な結果が得られなかった．そこで今後は，VirusTotal で取得できない項目について，独自の検体解析環境を用意して解析してランサムウェア検知に有効な特徴を調査する．

参考文献

[1] 独立行政法人 情報処理推進機構：情報セキュリティ 10 大脅威 2017, https://www.ipa.go.jp/security/vuln/10threats2017.html.

[2] Kharraz, A., Arshad, S., Mulliner, C., Robertson, W. K. and Kirda, E.: UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware., *USENIX Security Symposium*, pp. 757–772 (2016).

[3] Scaife, N., Carter, H., Traynor, P. and Butler, K. R.: Cryptolock (and drop it): stopping ransomware attacks on user data, *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, IEEE, pp. 303–312 (2016).

[4] VirusTotal: https://www.virustotal.com.

[5] VirusTotal: Private API v2.0, https://www.virustotal.com/en/documentation/private-api.

[6] NIST: SECURE HASH STANDARD, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

[7] VirusTotal: リファレンス，https://developers.virustotal.com/v2.0/reference.

[8] Microsoft: Cryptography (Windows) - MSDN, https://msdn.microsoft.com/ja-jp/library/windows/desktop/aa380255(v=vs.85).aspx.

[9] OpenSSL Software Foundation: Open SSL Cryptography and SSL/TLS Toolkit, https://www.openssl.org.

[10] Jeffrey Walton and the Crypto++ community: Crypto++ Library, https://www.cryptopp.com.

[11] Richter, J.: Load your 32-bit DLL into another process space using INJLIB, *Microsoft Systems Journal* (1994).

[12] SSDEEP: Private API v2.0, http://ssdeep.sourceforge.net.