マルウェアが実行する耐解析処理の定量的傾向

大山 恵弘

現代的なマルウェアの多くは解析を妨害するための処理（耐解析処理）を実行する．耐解析処理の例にはサンドボックスの検出や長時間のスリープの実行がある．耐解析処理を実現するための方法はよく研究されてきたが，現実のマルウェアが実行する耐解析処理の定量的傾向については，十分な知見が得られていない．本論文では，最近のマルウェアが実行する耐解析処理の定量的傾向を示す．例えば，どの程度の割合のマルウェアがどのような種類の耐解析処理を実行するかを明らかにする．

Many modern malware programs execute operations to hinder analysis (anti-analysis operations). Examples of them are sandbox detection and execution of long sleeps. Although the methods to achieve anti-analysis operations have been extensively studied, only insufficient knowledge has been obtained about the quantitative trends of anti-analysis operations executed by real malware. In this paper, we present quantitative trends of anti-analysis operations executed by recent malware. For example, we clarify what proportion of malware programs execute each particular class of anti-analysis operations.

1 はじめに

現代的なマルウェアの多くは，自身の解析を妨害するための処理（耐解析処理）を実行する．耐解析処理の例としては，自身が仮想マシンやサンドボックスなどの解析に用いられるシステム上で動いているかどうかの判定や，長時間のスリープの実行がある．

耐解析処理については多くの研究が行われ，耐解析処理の実現方法自体については多くの知見が集まっている．しかし，現在世界で観測されているマルウェアがどの程度の割合で耐解析処理を実行するのか，また，どんな種類の耐解析処理を実行するのかについては，知見が不足しており，明確にはなっていない．

そこで著者は，それらを明らかにする研究を行っている．例えば文献 [7,12] では，マルウェアの動的解析結果のデータセットである FFRI Dataset 2016 [13] に見られる，マルウェアが実行する耐解析処理の傾向に関する調査結果を示している．この調査結果には興味深い知見が多く含まれていると考えるが，最新のマルウェアが実行する広範な耐解析処理の傾向を必ずしも捉えてはいない．具体的には，データセットの収集に用いられた動的解析システム Cuckoo Sandbox [4] のバージョンが古かったために，解析結果に出力される耐解析処理の種類が限られていた．その制限は文献 [7,12] でも明示的に述べられている．

その後，新しい版である FFRI Dataset 2017 [11] が配布されるようになった．FFRI Dataset 2017 の収集に用いられた Cuckoo Sandbox は，FFRI Dataset 2016 のそれと比較して，マルウェアの特徴的な挙動を検出するためのより充実したルール集を備えており，はるかに多くの耐解析処理を検出できるようになっている．

本研究では，FFRI Dataset 2017 から観測できる，最新のマルウェアが実行する耐解析処理の傾向を示す．特に以下の点を明らかにする．

- どの程度の割合のマルウェアがどのような種類の耐解析処理を実行するか
- 耐解析処理によってどのようなシステムの検出が試みられているか

Quantitative Trends of Anti-Analysis Operations Executed by Malware.
Yoshihiro Oyama, 筑波大学, University of Tsukuba.

- 個々のマルウェアは何種類の耐解析処理を実行するか

本研究ではマルウェア解析のうち静的解析ではなく動的解析を主に扱う．また，耐解析処理についても，静的解析を妨害する処理ではなく動的解析を妨害する処理を主に扱う．

2 耐解析処理

耐解析処理の概要を述べる．耐解析処理の詳細は [1–3, 8–10] などの多くの文献で説明されている．主な耐解析処理を以下に示す．

解析システムの検出 自身がハイパバイザ，デバッガ，サンドボックスなどの解析用のシステム（解析システム）上で動いているかどうかを推定する．解析システム上で動いていると判断したら，実行終了などの解析を妨げる処理を行う．推定に用いられる手段としては，ハードウェア情報の検査，サービスやファイルの存在の検査，処理の所要時間やタイミングの測定，特定の CPU 命令の挙動の検査などがある．

解析時間の長大化 自身の重要な部分が実行されるまでに長大な時間がかかるようにする．動的解析ではしばしばタイムアウトが設定されるため，重要な部分の実行をそれ以降まで遅らせることによって，解析結果を無益なものにすることができる．実行を遅らせる手段としては，長時間のスリープの実行や無駄な処理の実行などがある．

解析処理の回避 自身に適用されている解析のための機構を検出し，その機構を無効化させる．例えば自身にセットされているデバッガのブレークポイントを外したり，解析システムのものと思われるプロセスを kill したりする．

3 FFRI Dataset

FFRI Dataset は Cuckoo Sandbox の上でマルウェアを実行して得られた動的解析結果のデータセットである．Cuckoo Sandbox は仮想マシンモニタを用いたサンドボックスであり，広く用いられている．そのソースコードは GitHub で公開されている．

本研究では FFRI Dataset 2017 に含まれる Windows 7 (x86) 上での 6251 検体のマルウェアの解析結果を用いる．これらの検体は 2017 年の 3 月から 4 月に FFRI 社が収集したものであり，どれも 15 以上のアンチマルウェア製品によってマルウェアと判定されている．データセット内の情報によると，データセットの作成には Cuckoo Sandbox 2.0.1 と VirtualBox が用いられた．各検体はタイムアウトを 90 秒に設定して実行された．

FFRI Dataset には多岐に渡る情報が含まれるが，本研究では以下の 2 つの情報を用いる．

- Windows API 呼び出しの列
- Signature

FFRI Dataset には他にも，マルウェアのバイナリに含まれる静的情報，マルウェアの実行中に発生した通信の情報，マルウェアがアクセスしたファイルやレジストリなどの資源の情報，公共サンドボックスによるスキャン結果などが含まれる．

API 呼び出し列の例を図 1 に示す．API 呼び出し列の情報はデータセットに JSON 形式で格納されているが，図にはそれを整形したものを示している．API 呼び出しの引数から，アクセスしたファイルやレジストリなどの資源の情報が得られる．

Signature は，バイナリや挙動に見られるマルウェアの特徴である．例えば，他プロセスへのコードインジェクション，HTTP 通信，実行型ファイルの生成，UPX によるパック，起動時に自動的に実行されるプログラムへの自身の登録などがある．Cuckoo Sandbox はマルウェアのバイナリや実行から signature を検出し，検出した signature 群をそのマルウェアの解析結果に記録する．例えば，Cuckoo Sandbox はファイル関連 API 呼び出しの引数を検査し，引数が “`VBoxHook.dll`” や “...`VirtualBox Guest Additions`...” などの VirtualBox を特徴づける文字列のパターンとマッチしたら，VirtualBox の検出を試みているという signature を解析結果に記録する．

4 分析結果

4.1 統計情報

FFRI Dataset 2017 の統計情報を表 1 に示す．各検体は平均で約 20000 回の API 呼び出しを実行して

```
...
NtCreateFile(filepath="\\??\\SIWVID", ...) = 3221225524 (STATUS_OBJECT_NAME_NOT_FOUND)
NtCreateFile(filepath="\\??\\NTICE", ...) = 3221225524 (STATUS_OBJECT_NAME_NOT_FOUND)
RegOpenKeyExA(regkey="HKEY_CURRENT_USER\\Software\\Wine", ...) = 2 (ERROR_FILE_NOT_FOUND)
LdrGetProcedureAddress(function_name="GetNativeSystemInfo", module="kernel32", ...) = 0
GetNativeSystemInfo(processor_count=...) = 0
GetSystemDirectoryA(dirpath=...) = 19
NtClose(0) = 3221225480 (STATUS_INVALID_HANDLE)
...
```

図 1　API 呼び出し列の例

表 1　統計情報

検体数	6251
各検体が実行する API 呼び出し数	最小: 95, 平均: 19993.1, 最大: 145234
各検体が使用する API 関数の種類数	最小: 14, 平均: 96.7, 最大: 181
全検体を通しての API 関数の種類数	292
各検体から検出された signature の数	最小: 4, 平均: 11.8, 最大: 27
全検体を通しての検出された signature の数	158
各検体のプロセスの数	最小: 1, 平均: 4.3, 最大: 55
各検体のスレッドの数	最小: 1, 平均: 20.7, 最大: 488

いる．各検体は平均で約 100 種類の API 関数を呼び出す．各マルウェアからは平均約 12 の signature が検出されている．詳細は後述するが，中には 27 の signature が検出されるような目立った挙動を示す検体も存在する．検出された signature の総数は 158 である．Cuckoo Sandbox 2.0.1 がリリースされた 2017 年 4 月 16 日時点での GitHub にあるソースコードによれば，signature（を検出するコードのファイル）の数は 379 である．すなわち，Cuckoo Sandbox が検出可能な signature の約 42%がこのデータセットに含まれている．プロセスの数とスレッドの数の平均はそれぞれ約 4.3 と約 20.7 であり，それほど大きくはないが，中には 55 のプロセスや 488 のスレッドで構成される検体も存在する．

4.2　多く検出された signature

各 signature が何検体から検出されたかを調査した．まず，全ての signature を対象に，検出された検体数が多い signature の上位 20 件を抽出した．次に，耐解析処理の signature（以下では単に耐解析処理 signature と呼ぶ）を対象に，上位 20 件を抽出した．Cuckoo Sandbox では各 signature を検出するコードは別々のファイルに記述されている．それらのうち `anti` という接頭語で始まるファイルが 52 あるが，これらのファイルのコードで検出される signature を，耐解析処理 signature とみなした．ファイル名の内訳は，`antianalysis_*` が 1，`antiav_*` が 5，`antidbg_*` が 2，`antiemu_*` が 1，`antisandbox_*` が 12，`antivirus_*` が 2，`antivm*` が 29 である．

検出された検体の数が多い signature の上位 20 件を表 2 に示す．検出された signature の総数は 158 なので，この並びは 158 位まで続く．最も多くの検体から検出された signature は，アンパックによく用いられる読み書き可能メモリを確保する挙動である．ただし，これは通常のプログラムの多くにも見られる挙動であり，それほどマルウェアに特徴的な挙動というわけではない．同様の微妙な挙動の signature は他にも多く存在する．以下の順位には，メモリ量のチェック，実行型ファイルの作成，バイナリ内の暗号化や圧縮がされたデータ，興味深いバッファの抽出，

表 2　検出された検体の数が多い signature の上位 20 件

	Cuckoo Sandbox による signature の説明	検体の数と割合
1	Allocates read-write-execute memory (usually to unpack itself)	5317 (85.1%)
2	Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available	4419 (70.7%)
3	Creates executable files on the filesystem	3872 (61.9%)
4	The binary likely contains encrypted or compressed data.	3795 (60.7%)
5	One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.	3660 (58.6%)
6	Queries for the computername	3610 (57.8%)
7	One or more processes crashed	3216 (51.4%)
8	Generates some ICMP traffic	2971 (47.5%)
9	Installs itself for autorun at Windows startup	2694 (43.1%)
10	Creates an Alternate Data Stream (ADS)	2494 (39.9%)
11	Creates a suspicious process	2493 (39.9%)
12	Drops a binary and executes it	2204 (35.3%)
13	Executed a process and injected code into it, probably while unpacking	2182 (34.9%)
14	Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)	2052 (32.8%)
15	The executable has PE anomalies (could be a false positive)	1807 (28.9%)
16	One or more of the buffers contains an embedded PE file	1703 (27.2%)
17	Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation	1423 (22.8%)
18	Creates (office) documents on the filesystem	1402 (22.4%)
19	This sample modifies more than 5 files through suspicious ways, likely a polymorphic virus or a ransomware	1340 (21.4%)
20	Attempts to detect Cuckoo Sandbox through the presence of a file	1261 (20.2%)

コンピュータ名の問い合わせ，マルウェアプロセスのクラッシュが続く．ここまでの signature は 50%以上の検体から検出されており，ある意味，平凡な挙動の signature と言える．耐解析処理 signature は 2 位，6 位，17 位，20 位に入っている．耐解析処理 signature は他の signature と比べて，それほど頻繁には検出されない．これは耐解析処理 signature の重要性を低下させるものではなく，むしろ逆である．耐解析処理 signature が検出された検体はそれだけで「目出つ」ことになり，他の検体との区別がしやすくなる．

検出された検体の数が多い耐解析処理 signature の上位 20 件を表 3 に示す．検出された耐解析 signature の総数は 40 なので，この並びは 40 位まで続く．上述したソースコードによれば，耐解析 signature（を検出するコードのファイル）の数は 52 なので，耐解析 signature のうち約 77%がこのデータセットに含まれている．

最上位には，耐解析処理のためにそれを実行したのかどうかが微妙であるような signature が並ぶ．1，2，3，7 位はメモリ量，コンピュータ名，ディスクサイズ，ネットワークアダプタのアドレスの検査であるが，それらは通常のプログラムも実行すると思われる．

特定の解析システムの検出に関する signature も見られる．具体的には Cuckoo Sandbox，VMware，VirtualBox，Avast，Sunbelt Sandbox などの製品に関する signature が見られる．21 位以下には ThreatTrack，GFI SandBox，CWSandbox, Wine，VirtualPC，Sandoxie，BitDefender，Xen に関する signature が出現する．特定の解析システムのうち，最も多くの検体で検出が試みられたと判定されたのは Cuckoo Sandbox であり，続いて VMware，VirtualBox，Sunbelt Sandbox である．広く普及しているシステムが上位に来ている．これら 4 つ以外の製品を検出しようとする signature が検出されることは稀であり，高々1%程度の検体からしか検出されていない．

6 位と 7 位はそれぞれ，スリープによる解析時間の長大化とフォアグラウンドウィンドウの検査による人

表 3　検出された検体の数が多い耐解析処理の signature の上位 20 件

	Cuckoo Sandbox による signature の説明	検体の数と割合
1	Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available	4419 (70.7%)
2	Queries for the computername	3610 (57.8%)
3	Queries the disk size which could be used to detect virtual machine with small fixed size or dynamic allocation	1423 (22.8%)
4	Attempts to detect Cuckoo Sandbox through the presence of a file	1261 (20.2%)
5	A process attempted to delay the analysis task	867 (13.9%)
6	Checks whether any human activity is being performed by constantly checking whether the foreground window changed	643 (10.3%)
7	Checks adapter addresses which can be used to detect virtual network interfaces	436 (7.0%)
8	Detects VMWare through the in instruction feature	374 (6.0%)
9	Detects VirtualBox through the presence of a file	346 (5.5%)
10	Detects Avast Antivirus through the presence of a library	339 (5.4%)
11	Detects VMWare through the presence of various files	313 (5.0%)
12	Looks for the Windows Idle Time to determine the uptime	310 (5.0%)
13	Attempts to identify installed analysis tools by a known file location	300 (4.8%)
14	Detects Sunbelt Sandbox through the presence of a file	297 (4.8%)
15	Checks for the presence of known windows from debuggers and forensic tools	249 (4.0%)
16	Attempts to identify installed AV products by registry key	164 (2.6%)
17	Enumerates services, possibly for anti-virtualization	149 (2.4%)
18	Checks for known Chinese AV sofware registry keys	135 (2.2%)
19	Attempts to identify installed AV products by installation directory	121 (1.9%)
20	Checks for the presence of known devices from debuggers and forensic tools	90 (1.4%)

の活動の検出であり，ここまでが 10%以上の検体から検出された signature である．

検出される signature の傾向は FFRI Dataset 2016 での傾向 [7, 12] から大きく変わっている．例えば，表 3 の 1，2，3，4，7，8，10，13，14，16，19，20 位は，FFRI Dataset 2016 では 1 回も検出されていない signature である．全体的には，検出される signature が増える方向に変化している．変化は Cuckoo Sandbox のバージョンアップとマルウェアの進化の両方に起因すると考えるが，少なくとも前者の影響は大きいと推測している．バージョンアップにより，新しい signature が検出できるようになっているとともに，4.4 節で述べるように，signature 検出において見逃されていた多くの API 呼び出しの引数が見逃されなくなっている．その結果，FFRI Dataset 2017 では FFRI Dataset 2016 と比べて耐解析処理の実行の実態をより詳しく把握できるようになっている．

20.2%もの検体から Cuckoo Sandbox に関する signature が検出されているが，これは誤検出を含んでいる可能性がある．マルウェアが Cuckoo Sandbox を検出しようとしているかどうかは，ファイル関連 API 呼び出しの引数に，所定のパターンにマッチする文字列が与えられたかどうかなどの条件で判定される．例えばマルウェアが `agent.py` や `analyzer.py` というファイルを open したら，Cuckoo Sandbox を検出しようとしたとみなされる．それらは通常 Cuckoo Sandbox の上で動くゲスト OS に置かれるファイルである．マルウェアは耐解析処理以外の目的で，ルートフォルダ，ユーザの Document フォルダ，ユーザの Recent フォルダなどにある全ファイルを open することがある．例えばランサムウェアが暗号化対象ファイルを探す目的でそれを実行する．その際，それらのフォルダに `agent.py` や `analyzer.py` があれば，それらが open される．このように，Cuckoo Sandbox の検出を試みていないのに試みたと判定された検体が，データセットに一定数含まれていると予想している．実際，Cuckoo Sandbox の signature が検出された検体のうちの多くで，特徴的なフォルダ内の全ファイルのスキャンの結果として `agent.py` や `analyzer.py` がアクセスされたと思わせる API 呼び出し列が実行

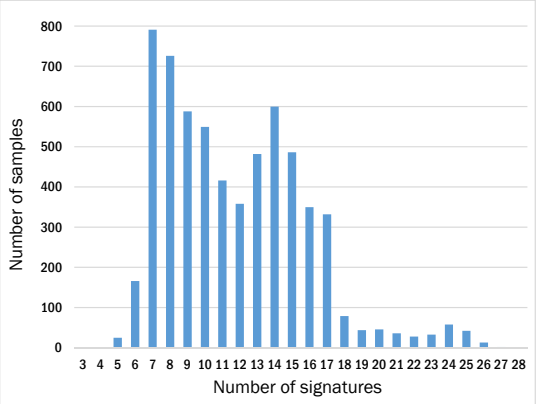

図 2　検出された signature の数ごとの検体数

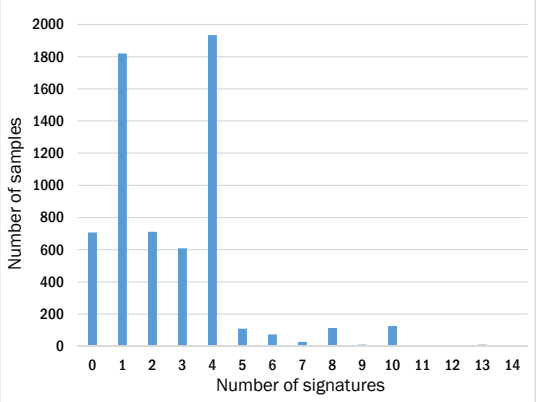

図 3　検出された耐解析 signature の数ごとの検体数

されている．現状では，Cuckoo Sandbox はかなり単純な引数のパターンマッチに基いて signature を検出しており，マルウェアが本当に耐解析処理を実行しようとしたのかどうかを詳しく検査していないという点には注意が必要である．

4.3　検出された signature の数

Signature の数ごとの検体数の分布を調査した．全 signature についての分布を図 2 に，耐解析処理 signature についての分布を図 3 に示す．

1 検体から検出された signature の数は 4 から 27 の範囲内にある．6 から 17 の signature が検出された検体の数はいずれも 100 を越えており，この範囲の数の signature を実行する検体が多い．5 以下および 18 以上の signature が検出される検体は少ない．ただし，表 1 で示したように，最多の 27 の signature が検出された検体も存在する．

検出された耐解析処理 signature はずっと少なくなっている．1 検体から検出された signature の数は 0 から 13 の範囲内にある．その数が 4 以下だった検体が多く，5 以上だった検体は全体の約 7.5%である．その数の平均は約 2.6 である．すなわち，広範な耐解析処理を実行するマルウェアは少ない．ただし，最多の 13 の耐解析処理 signature が検出された検体も存在する．

27 の signature かつ 13 の耐解析処理 signature が検出された検体がデータセットに 1 検体だけ存在する．この検体から検出された signature を図 4 に示す．特定のシステムに関する耐解析 signature として，VirtualBox，VMware，VirtualPC，Wine に関する signature が検出されている．他にも，耐解析処理 signature としては，様々な資源の状態や属性の検査，解析時間の長大化などが検出されている．広範な耐解析処理を実行する洗練されたマルウェアも，数は少ないながら存在することがわかる．

4.4　制限

Cuckoo Sandbox が検出する signature には当然ながら false positive と false negative が含まれる．詳しくは文献 [7, 12] に述べられているが，それらが生じる理由としては，まず，検査すべき API 関数や照合すべき文字列のパターンが網羅されていないというものがある．そもそも全てのパターンを網羅するのはほぼ不可能である．FFRI Dataset 2016 では加えて，一部の API 関数（`NtCreateFile` など）で，失敗した呼び出しの引数は検査対象から外されており，それが大量の false negative をもたらしていた．この制限は FFRI Dataset 2017 では無くなっており，検出される signature が大幅に増えている．さらに前述のように，フォルダ内の全ファイルの open をマルウェアが実行した場合，解析システムを検出しようとしてはいないのに，解析システムを特徴づけるファイルを open することになり，解析システムを検出しようとしているとみなされることがある．それは false positive に

```
耐解析 signature
- Queries for the computername
- Checks amount of memory in system, this can be used to detect virtual machines that have a
  low amount of memory available
- A process attempted to delay the analysis task.
- Looks for the Windows Idle Time to determine the uptime
- Checks the version of Bios, possibly for anti-virtualization
- Detects virtualization software with SCSI Disk Identifier trick(s)
- Detects VirtualBox through the presence of a file
- Detects VirtualBox through the presence of a registry key
- Tries to detect VirtualPC
- Detects VMWare through the presence of various files
- Detects VMWare through the presence of a registry key
- Detects Virtual PC through the presence of a registry key
- Detects the presence of Wine emulator

他の signature
- Collects information to fingerprint the system (MachineGuid, DigitalProductId, SystemBiosDate)
- This executable has a PDB path
- One or more potentially interesting buffers were extracted, these generally contain injected
  code, configuration data, etc.
- Allocates read-write-execute memory (usually to unpack itself)
- Creates executable files on the filesystem
- Creates a suspicious process
- Tries to locate whether any sniffers are installed
- The binary likely contains encrypted or compressed data.
- One or more of the buffers contains an embedded PE file
- Installs itself for autorun at Windows startup
- Attempts to modify browser security settings
- Generates some ICMP traffic
- Executed a process and injected code into it, probably while unpacking
- Connects to IP addresses that are no longer responding to requests
  (legitimate services will remain up-and-running usually)
```

図 4　単一の検体から検出された 27 の signature

つながる．本研究の結果は，Cuckoo Sandbox による signature 検出の特徴を十分に理解した上で見ることが望ましい．ただ，一般にはマルウェアの意図を API 呼び出し列から常に正確に判断することは極めて難しく，どのシステムやデータセットを用いても，同様の不正確さは生じると考えている．

5　関連研究

文献 [1,2] の研究でも本研究と同様に，現代的なマルウェアのどの程度の割合が耐解析処理を使用するかを明らかにしている．しかし，彼らの研究は本研究とは異なり，静的解析を用いている．静的解析で捉えられる特徴と動的解析で捉えられる特徴は異なるため，両方の解析手法を補完的に用いて知見を蓄積することが重要である．

BareCloud についての研究 [6] では，マルウェア 110005 検体のうち 5835 検体が，耐解析処理などを実行する evasive なマルウェアであったことが報告されている．彼らの研究は本研究とは異なり，マルウェアが具体的にどんな耐解析処理を実行したかを明らかにしていない．

Ferrand [5] は Cuckoo Sandbox を検出するための複数の方法およびその検出を回避するための方法を示している．彼らの研究は耐解析処理の実現方法およびその回避方法を示す研究であり，本研究は，そのような耐解析処理の使用の傾向を示す研究である．

SandPrint [10] は，サンドボックス内で動くマルウェアがそのサンドボックスの特徴を抽出し，それがどのサンドボックスであるかを高い精度で特定することを可能にするシステムである．SandPrint は広範な耐解析処理の実行結果を組み合わせて精度を高めている．SandPrint は高度な耐解析処理によって実現可

能な脅威を示す研究であり，本研究はマルウェアによる耐解析処理の使用の実態を示す研究である．

AADetect [9] は，耐解析処理の特徴の有無に基いてプログラムがマルウェアかどうかを判定するシステムである．AADetect の研究は耐解析処理に関するマルウェアと非マルウェアの挙動の違いを明らかにしているが，現実のマルウェアの多くが実行する耐解析処理や，各耐解析処理を実行するマルウェアの割合を示していない．実験で用いているマルウェアは 69 検体であり，本研究よりも大幅に少ない．

6 まとめと今後の課題

2017 年に収集されたマルウェアが実行する耐解析処理の傾向を，API 呼び出し列の分析によって示した．最も多くの検体で検出された耐解析 signature は，メモリ量，コンピュータ名，ディスクサイズの検査である．特定のシステムに関する signature としては，20%以上の検体の実行から，Cuckoo Sandbox を検出しようとしているという signature が検出された．他には，VMware，VirtualBox，Avast，Sunbelt Sandbox に関する signature が多く検出された．他に顕著だった耐解析 signature は，スリープによる解析時間の長大化，フォアグラウンドウィンドウの検査による人の活動の検出，ネットワークアダプタのアドレスの検査である．マルウェア 1 検体から検出された signature の数の平均は約 11.8 であり，耐解析 signature に限ると約 2.6 であった．ただし，27 もの signature や 13 もの耐解析 signature が検出された検体もあった．

今後の課題を述べる．まず，API 呼び出し列をさらに詳しく分析し，耐解析処理の実行後にマルウェアがどう振る舞ったかを解明する必要がある．特に，サンドボックスの検出に伴い実行を終了させる処理を実際に行ったかどうかを調べる必要がある．また，現在の Cuckoo Sandbox による signature 検出では少なくない false positive や false negative が出ると思われるため，それらの誤りの発生を減らす技術や，それらの誤りを解析結果から発見，除去する技術が必要である．

謝辞 FFRI Datasets を提供して下さった株式会社 FFRI および MWS 組織委員会に感謝する．本研究の一部は JSPS 科研費 17K00179，26330080 の助成を受けている．

参 考 文 献

[1] Barbosa, G. N. and Branco, R. R.: Prevalent Characteristics in Modern Malware, Black Hat USA 2014, 2014.

[2] Branco, R. R., Barbosa, G. N., and Neto, P. D.: Scientific but Not Academical Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies, Black Hat USA 2012, 2012.

[3] Chen, X., Andersen, J., Mao, Z. M., Bailey, M., and Nazario, J.: Towards an Understanding of Anti-virtualization and Anti-debugging Behavior in Modern Malware, *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2008, pp. 177–186.

[4] Cuckoo Sandbox, `https://cuckoosandbox.org/`.

[5] Ferrand, O.: How to detect the Cuckoo Sandbox and to Strengthen it?, *Journal of Computer Virology and Hacking Techniques*, Vol. 11, No. 1(2015), pp. 51–58.

[6] Kirat, D., Vigna, G., and Kruegel, C.: BareCloud: Bare-metal Analysis-based Evasive Malware Detection, *Proceedings of the 23rd USENIX Security Symposium*, 2014, pp. 287–301.

[7] Oyama, Y.: Trends of anti-analysis operations of malwares observed in API call logs, *Journal of Computer Virology and Hacking Techniques*, (2017).

[8] Raffetseder, T., Kruegel, C., and Kirda, E.: Detecting System Emulators, *Proceedings of the 10th Information Security Conference*, 2007, pp. 1–18.

[9] Tan, J. W. and Yap, R. H.: Detecting Malware Through Anti-analysis Signals - A Preliminary Study, *Proceedings of the 15th International Conference on Cryptology and Network Security*, 2016, pp. 542–551.

[10] Yokoyama, A., Ishii, K., Tanabe, R., Papa, Y., Yoshioka, K., Matsumoto, T., Kasama, T., Inoue, D., Brengel, M., Backes, M., and Rossow, C.: SandPrint: Fingerprinting Malware Sandboxes to Provide Intelligence for Sandbox Evasion, *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses*, 2016, pp. 165–187.

[11] 株式会社 FFRI: FFRI Dataset 2017 のご紹介, MWS2017 意見交換会, 2017.

[12] 大山恵弘: マルウェアによる対仮想化処理の傾向についての分析, コンピュータセキュリティシンポジウム 2016 論文集, 2016, pp. 534–541.

[13] 高田雄太, 寺田真敏, 村上純一, 笠間貴弘, 吉岡克成, 畑田充弘: マルウェア対策のための研究用データセット〜MWS Datasets 2016〜, 情報処理学会研究報告 コンピュータセキュリティ, Vol. 2016-CSEC-74, 2016.