Vol.2018-IOT-43 No.11
Vol.2018-SPT-30 No.11
2018/9/28

API 間の相関性に基づくランサムウェア亜種を区別する提案

周家興　廣瀬幸†　柿崎淑郎†　猪俣敦夫†

概要：ランサムウェアが実行される時，ファイルを削除したり、暗号化したり、または他の動作を行うために、必ず API を使う．更に、同じファミリーに属するランサムウェア亜種であれば、実行された時ランサムウェアの親プロセスに使用された API の種類が同じである．本稿ではランサムウェアが実行された時の親プロセスにより呼び出された API の頻度に着目し、API 同士間の相関係数を求め、その相関係数を特徴量として機械学習を用いてランサムウェア亜種のファミリーを区別する方法を提案する．

キーワード：API 頻度，相関係数，機械学習，亜種

Proposal to Distinguish between Ransomware Variants based on Correlation between APIs

Jiaxing Zhou　Miyuki Hirose†　Yoshio Kakizaki†　Atsuo Inomata †

Abstract: When Ransomware is run, be sure to use the API to delete files, encrypt them, or perform other actions. Furthermore, if it is a Ransomware subspecies belonging to the same family, the type of API used for the parent process of Ransomware when executed is the same. In this paper, we focus on the frequency of API called by the parent process when Ransomware is executed, find the correlation coefficient between APIs, and we propose a method to distinguish family of Ransomware subspecies using machine learning with its correlation coefficient as feature quantity.

Keywords: API frequencies, correlation coefficient, machine learning, subspecies

1. はじめに

近年、ランサムウェアの数を迅速に増やしていく伴い、ランサムウェアによる被害事件も年々増加していく．特に 2016 年は国内での検出台数、被害報告件数が過去最多となった[1]．また、株式会社フォーティネットジャパンの調査により、2016 年の毎日 4000 件以上のランサムウェア攻撃が発生し、感染デバイス台数は、平均で毎月 30000~50000 台にも達している[2]．日本だけではなく、中国の国立インターネット緊急センター（CNCERT/CC）の「2017 年中国インタネットサイバーセキュリティレポート」により、2017 年での新しい種類のランサムウェア検出台数が 40000 台を超えた[3]. 2017 年での検出されたマルウェア種類数の中に、ランサムウェアの種類数が 2 位になり、12.25%になった[4]. その中、近年見られるランサムウェア亜種の数の増加があり、特に Windows 関連の亜種ファミリーが検知された数が毎年伸びている[5]．ランサムウェアの検知をよく研究されているが、ランサムウェア亜種の検知に対し，従来の検知手法が対応しにくくなってくる．

しかし，どんな種類のランサムウェアであっても、システムの API を呼び出す．それらが同じファミリーに属している限り，さまざまな亜種により呼び出された API シーケンスの間にいくつかの類似点が存在する[6]．したがって，同じファミリーに属する各亜種により呼び出された API シーケンスのうちに、API 種類の間にある関係が存在し、その結果、API シーケンスの間にある程度の類似性が生じる．本稿では，ランサムウェアにより呼び出された API の頻度に着目し、各 API 間の相関係数求め、機械学習を用いてランサムウェア亜種のファミリーを区別する方法について提案する．

本稿の提案手法によって、ランサムウェアファミリーを区別できるだけではなく、ランサムウェアファミリーの API シーケンスを確定できるだと考えている．

2. 関連研究

ランサムウェアを検知するために、呼び出された API、DLL の頻度、特定種類のファイルに対する削除、変更等動作の頻度およびメモリ特徴に基づき、ランダムフォレストを用いて、ランサムウェアを検知する方法を Gong Qi らが提案した[7]．Gong Qi らの提案手法が従来の検知手法と比べて、より効果的にランサムウェアとその亜種を検知できるだけど、検知手法の時間計算量が高い．

マルウェアにより呼び出された API の頻度を用いてアンサンブル学習を使用し、マルウェアの検知手法を Pratiksha Natani らが提案した[8]．この研究の提案手法で悪意のある行動とその API 関数を定義したが、種類数数が限られて、汎用性を改善する必要性がある．In Kyeom Cho らがマルウェアにより呼び出された API シーケンスをアライメントし

† 東京電機大学，東京都
Tokyo Denki University, Tokyo Senju campus, 5 Senju Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan.

て、アライメントされたシーケンス間の類似度を用いて、同じマルウェアファミリーのサンプル間の類似度が高く、異なるファミリーのサンプル間の類似度が低いことことを証明した．マルウェアにより呼び出された DLL ファイル、API、API 関数のパラメータの類似度に基づきマルウェアを検知する手法を Z.Salehi らに提案された[9]．しかし、頻度が閾値を超えるとその API を選ぶことであれば、適切な閾値の決め方が重要である．

関連研究によって、API シーケンスとその頻度を特徴量として使用したらより効果的にマルウェアを検知できるため、本稿で API 種類間の相関性を用いてランサムウェア亜種を区別するのが従来の検知手法と比べて汎用性が高いと考えている．

3. 提案手法

近年、いろんな新たな種類のランサムウェアが出てくるのはマルウェアの数が年々増やしている原因の一つであるが、もう一つはランサムウェア亜種の数が増加していくことである．ランサムウェア亜種とは、特定のランサムウェアに似ていながら、微妙に異なる特徴を持つもの．ランサムウェア亜種を検出するために、本研究で提案される手法は API 間の相関性に基づく機械学習を用いてランサムウェアファミリーを区別することである．Gong Qi らの提案手法で API 頻度を特徴量として使用してランサムウェアの検知には有効性があるではなく、I. K. Cho らの研究結果から API シーケンス間に類似性があるによって、API 種類の間にある程度の関係が存在する可能性があるため、本稿では API 頻度と API 間の相関性を使用する．つまり、同じファミリーに属するランサムウェアの親プロセスの API 間の相関性をパターンとして使用し、分類を行う．その中、相関係数が実験の精度と関係があるため、API 種類を選択することが重要である．また、本文の提案手法に使用される特徴量が既存の検知手法より少なく、使用される特徴が亜種の分類することにもっと有効性を持つだと考えている．実験の流れが図 1 提案手法の概要図の通りである．ランサムウェア検体サンプルを Cuckoo sandbox に実行させて、解析を終わって analysis report の中にある report.json ファイルからエクセルで apistats を抽出する．そして、エクセルの CORREL 関数を用いて API 間の相関係数を計算する．相関性に対し、相関係数とは二つの確率変数の間にある線形な強弱を測る指標である．

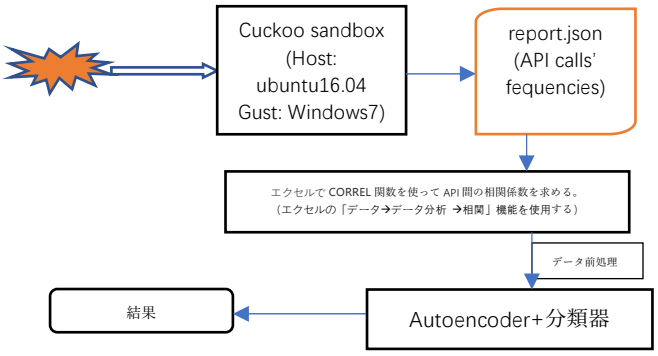

図 1 提案手法の概要図

4. 実験

1) 実験目的

サンドボックスを使用して各ランサムウェアファミリーの API シーケンスを抽出し、それらの API シーケンスを特徴量として使ってランサムウェア亜種を分類することである．

2) 実験方法

まず、実験環境については、Cuckoo Sandbox[10]を使用する．Cuckoo Sandbox に疑わしいファイルを投げ入れて、数分で実行させてから、ファイルの動作を概説した詳細なレポートをもらえる．そのレポートの中に、サンプルにより呼び出された API の種類、API 頻度を記載されている．

実験環境を図 2 で示す．

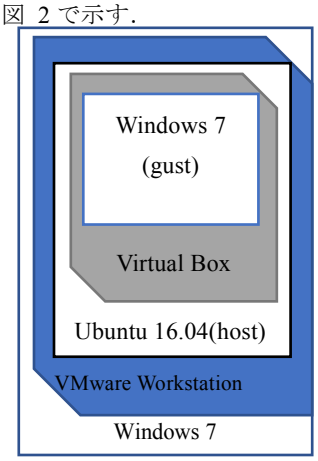

図 2 実験環境の構成

次、データセットと特徴量については、本研究でランサムウェア亜種を判定しに使用する API の種類がファイルを開く、削除し、コピーする API、レジストリを操作する API とプロセスに関する API などである．また、同じファミリーに属するランサムウェア亜種が実行する時にプロセスツリーがあるので、同じファミリーに属するものであれば、プロセスツリーの親プロセスの API の種類が同じであるはずことにより、本研究で親プロセスにある API 種類と各種類の頻度を特徴量として使用する．

実験で使用されたランサムウェアファミリーの種類が表 1 ランサムウェア種類である．

ファミリー
Locky

CryptoLocker
Cerber
Jigsaw
CryptoWall

表 1 ランサムウェア種類

ランサムウェア検体サンプルの数が不足のため、本文で表 1 にあるランサムウェアを使用して説明する.

最後、実験の手順を説明する. ランサムウェアが実行される時、プロセスツリーが一つだけではない可能性があるので、本研究で、プロセスの数が一番多いプロセスツリーを使用する.

Cuckoo sandbox から解析レポート report.json ファイルをもらって, エクセルの「データ取得」を使用し、report.json ファイルをインポートする.

例（Cerber の検体、プロセスが 17 個ある）としては

図 3 process tree のように, 長方形にある数字がプロセスの id（pid と略する）. 一番上の数字が 2448 であるが, 図 4 process id and record により、親プロセス id は 2472 であることが分かる.

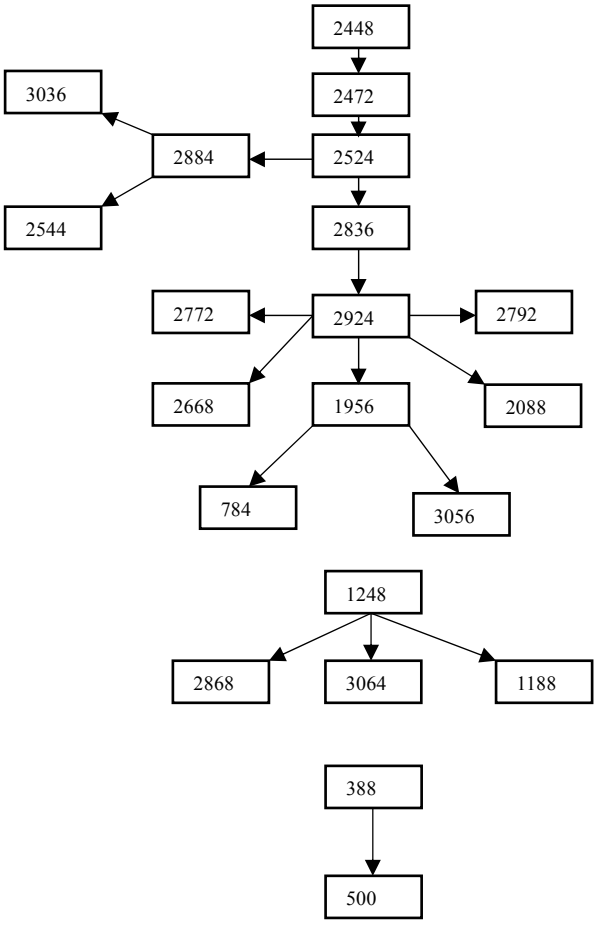

図 3 process tree

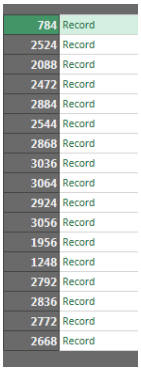

図 4 process id and record

図 4 process id and record にある record の中に API 種類と API の頻度を記載されている.

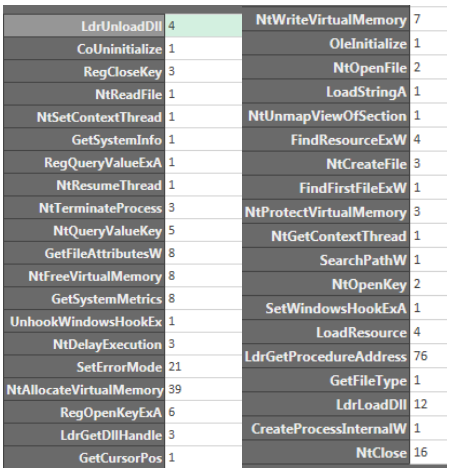

図 5 API シーケンス及び頻度

pid が 2472 であるプロセスの API シーケンスと頻度を図 5 で示す. シーケンスが長いため、2 列で表す.

この手順の通り、ランサムウェア検体の解析レポートから API シーケンスと API 頻度を抽出する.

3) 実験結果

ランサムウェア検体の数が少ないため、本文の結果に各ランサムウェアファミリーに共通である API シーケンスを示す.

また、実験で収集した各ランサムウェアファミリーの API シーケンスが下記の内容だけではなく、実験結果からサンプルのファミリーの所属を確定できるが、プロセスの数と親プロセスに含まれる API 種類が同じではないサンプルがあるため、本章で各ファミリーの同じ親プロセス API 種類を持つサンプルの API シーケンスのみを示す. 原因としては、違う親プロセスの API 種類を持つサンプルがあるため、例を挙げて説明する. ランサムウェア Cerber の解析レポートから、プロセスの数が 1、2、3、5、7、17 であるのサンプルがあり、それらのサンプルを Cerber として確定した根拠は Virustotal のレポートと実験の screenshot である. その中、7 個プロセスを持つサンプルの数が一番多いため、

7 個プロセスを持つサンプルの親プロセスの API シーケンスを結果として示す．Cerber 以外のランサムウェアもこのやり方でデータを収集する．

実験で解析されたランサムウェア種類を表 1 で示すように，各ファミリーの API シーケンスと頻度が以下のとおりである．左の列は API の名前を示し、右の列は各 API の頻度を示す．

まず、Locky ファミリーの API シーケンスと頻度を表 2 で示す．

LdrUnloadDll	1
NtReadFile	2
CreateActCtxW	4
NtOpenKey	1
SetFilePointer	1
SetUnhandledExceptionFilter	1
SetErrorMode	1
NtCreateFile	1
NtClose	6
GetSystemTimeAsFileTime	1
LdrLoadDll	1
NtTerminateProcess	3
GetFileAttributesW	2
CreateProcessInternalW	1
NtQueryValueKey	1

表 2 Locky ファミリーの API 頻度

API シーケンスが一番短いのは Locky の API シーケンスであり、その中、主な API はファイル、レジストリとプロセスに対する操作 API であることがわかった．この API シーケンスに暗号化する API が出なかったが、別の Locky ファミリー属するサンプルの親プロセス API シーケンスの中に暗号化する API が出た．そのサンプルが Locky であることを確定できるが、親プロセスの API シーケンスは表 2 の内容と違う．

CryptoLocker の API シーケンスが長いため、表 3 に 2 列で示す. CryptoLocker の API シーケンスの中に、頻度が 1000 回を超えた API が幾つかあり、ほぼファイルに対する操作である．この中、暗号化する API も出てきたので、頻度が 1000 回を超える原因はフィアルを暗号化するため、まずファイルのパスを探し、コピーしてから暗号化を行い、もとのファイルを削除するだと考え、またファイルの数と関係あるだと考えることである．また、レジストリだけではなく、カーネルに関する API も幾つか出てきて、特徴として使用できるだと考えている．

NtOpenSection	2	LdrLoadDll	63
getaddrinfo	1	NtEnumerateValueKey	506
CryptEncrypt	3	UuidCreate	1
GetFileVersionInfoSizeW	1	GetNativeSystemInfo	1
GetFileAttributesW	16	RegCloseKey	84
VolumePathNamesForVolumeNar	8	NtDuplicateObject	8
NtReadVirtualMemory	110	NtCreateKey	5
RegOpenKeyExW	132	WSAStartup	1
NtDelayExecution	414	CryptCreateHash	4
SetErrorMode	88	GetSystemMetrics	78
ShellExecuteExW	1	GetFileSize	1319
RegOpenKeyExA	2	DrawTextExW	26
GetCursorPos	4	NtAllocateVirtualMemory	1441
RtlAddVectoredExceptionHandle	1	GetComputerNameW	1
FindResourceExW	2	NtResumeThread	4
NtCreateFile	1342	SHGetFolderPathW	7
GetSystemTimeAsFileTime	39	RegEnumKeyExW	27
GlobalMemoryStatusEx	1	CryptAcquireContextW	2
LoadResource	1	closesocket	2
CoInitializeSecurity	1	RegEnumValueW	94
SetFileAttributesW	1	NtCreateSection	894
NtQueryInformationFile	1313	GetTimeZoneInformation	2
RegCreateKeyExW	3	SearchPathW	464
NtOpenMutant	2	GetFileType	2
NtQueryKey	42	CreateProcessInternalW	2
IsDebuggerPresent	1	WSASocketW	2
LookupPrivilegeValueW	1	LdrUnloadDll	5
NtQueryValueKey	35	GetShortPathNameW	1
RegQueryValueExW	201	GetSystemInfo	102
CreateActCtxW	5	setsockopt	2
NtDeviceIoControlFile	1	GetSystemWindowsDirectoryW	9
NtReadFile	25	NtClose	2320
LdrGetDllHandle	89	NtOpenProcess	5
NtQuerySystemInformation	1	CryptHashData	2
NtSetValueKey	1	NtFreeVirtualMemory	1254
CreateThread	6	RtlAddVectoredContinueHandler	1
GetSystemDirectoryW	8	NtMapViewOfSection	894
SetUnhandledExceptionFilter	1	NtOpenFile	21
VolumeNameForVolumeMountPo	3	RegQueryInfoKeyW	3
NtProtectVirtualMemory	87	NtUnmapViewOfSection	884
CoInitializeEx	4	NtQueryDirectoryFile	21
RegSetValueExW	4	NtQueryAttributesFile	5
LoadStringW	5	NtCreateMutant	17
LdrGetProcedureAddress	572	GetFileAttributesExW	14
CopyFileW	1	NtOpenKey	12
NtOpenDirectoryObject	1	GetFileVersionInfoW	1
		NtOpenKeyEx	34

表 3 CryptoLocker ファミリーの API 頻度

Cerber ファミリーの API シーケンスと頻度を表 4 で示す．

NtDuplicateObject	5	SetFilePointer	18
NtOpenSection	6	OleInitialize	1
CoUninitialize	16	NtOpenFile	3
RegCloseKey	368	GetFileSizeEx	12
NtReadFile	136	NtUnmapViewOfSection	11
GetNativeSystemInfo	1	CoCreateInstance	2
NtSetContextThread	1	GetSystemDirectoryW	2
SetFilePointerEx	1	NtQueryDirectoryFile	110
RegQueryValueExA	2	LoadStringW	1
NtResumeThread	1	GetTempPathW	1
GetSystemWindowsDirectoryW	8	NtCreateFile	75
NtQueryValueKey	10	VolumeNameForVolumeMountPo	4
FindResourceExW	2	GetSystemTimeAsFileTime	82
NtOpenProcess	14	GlobalMemoryStatusEx	1
GetFileAttributesW	27	NtQueryAttributesFile	2
RegQueryValueExW	2244	FindFirstFileExW	5
NtMapViewOfSection	9	NtCreateMutant	2
VolumePathNamesForVolumeNar	8	NtProtectVirtualMemory	1
RegEnumKeyW	111	CoInitializeEx	16
CreateActCtxW	2	GetFileInformationByHandleEx	12
GetFileSize	2	NtCreateSection	7
WriteProcessMemory	6	NtGetContextThread	1
RegOpenKeyExW	483	NtOpenKey	30
NtDelayExecution	1	LoadResource	2
SetErrorMode	9	LdrGetProcedureAddress	64
NtAllocateVirtualMemory	38	CreateDirectoryW	14
ReadProcessMemory	1	RtlDecompressBuffer	1
RegOpenKeyExA	6	SetFileTime	5
DeleteFileW	3	LdrLoadDll	17
NtWriteFile	30	NtTerminateProcess	3
LdrGetDllHandle	17	NtQueryInformationFile	3
NtFreeVirtualMemory	58	CreateProcessInternalW	1
LdrUnloadDll	3	NtClose	210
NtQuerySystemInformation	1		

表 4 Cerber ファミリーの API 頻度

Cerber の API シーケンス中に、ファイルの作成、コピー、削除等 API、レジストリに関する API などほぼあり，前述の通り、違う API シーケンスを持つサンプルもあるが、共通点として、レジストリに対する操作の API 回数が 2000 回を超えた．例としては、表 4 にある RegQueryValueExW だ．

Jigsaw ファミリーの API シーケンスと頻度を表 5 で示す．

RegCreateKeyExW		1	NtOpenFile		17
LdrUnloadDll		7	SHGetFolderPathW		5
DeviceIoControl		1	CreateThread		3
RegCloseKey		75	NtUnmapViewOfSection		9
NtQueryKey		32	RegQueryInfoKeyW		3
NtDuplicateObject		6	GetSystemDirectoryW		5
GetShortPathNameW		1	RegEnumKeyExW		21
GetSystemInfo		76	NtQueryDirectoryFile		19
CoInitializeEx		2	SetUnhandledExceptionFilter		1
LoadStringW		1	NtCreateFile		21
NtResumeThread		2	GetVolumeNameForVolumeMou		3
GetSystemWindowsDirectoryW		8	GetSystemTimeAsFileTime		23
NtClose		152	GlobalMemoryStatusEx		2
GetFileVersionInfoSizeW		1	NtQueryAttributesFile		1
NtOpenProcess		2	NtCreateMutant		5
GetFileAttributesW		7	NtProtectVirtualMemory		60
RegQueryValueExW		156	GetFileAttributesExW		21
NtFreeVirtualMemory		79	RegEnumValueW		5
GetVolumePathNamesForVolum		8	NtCreateSection		7
RtlAddVectoredContinueHandler		1	RegSetValueExW		1
NtMapViewOfSection		7	NtOpenKey		13
RtlAddVectoredExceptionHandle		1	NtOpenMutant		1
CreateActCtxW		2	LdrGetProcedureAddress		334
GetFileSize		5	CoInitializeSecurity		1
RegOpenKeyExW		98	CreateDirectoryW		3
NtDelayExecution		10	CopyFileW		2
SetErrorMode		70	GetFileVersionInfoW		1
ShellExecuteExW		1	IsDebuggerPresent		1
NtAllocateVirtualMemory		174	NtOpenDirectoryObject		1
RegOpenKeyExA		2	NtOpenKeyEx		30
DeleteFileW		3	LdrLoadDll		39
LdrGetDllHandle		136	NtTerminateProcess		3
GetNativeSystemInfo		1	UuidCreate		1
NtQuerySystemInformation		3	CreateProcessInternalW		1
NtReadFile		10	NtQueryValueKey		37

表 5 Jigsaw ファミリーの API 頻度

Jigsaw の API シーケンスに新しい API（DeviceIoControl）が出た．他のランサムウェアファミリーにたまにあるが、Jigsaw ファミリーの過半数の API シーケンスに DeviceIoControl がある．また、他のランサムウェアと同じ、ファイル、レジストリ等の API もある．

最後、CryptoWall ファミリーの API シーケンスと頻度を表 6 で示す．

CryptoWall サンプルの API シーケンスがほぼ短く、ここで使用した API シーケンスに、API 種類が他のランサムウェアの API シーケンスと違うが、ほぼファイル、レジストリなどの API であることがわかった．

CoInitializeEx		65
CoUninitialize		56
NtResumeThread		12
NtOpenFile		266
NtWriteFile		63
CoCreateInstance		53
CoInitializeSecurity		1
NtDelayExecution		39
URLDownloadToFileW		2
NtCreateFile		349
NtAllocateVirtualMemory		108
LdrLoadDll		133
OleInitialize		2
CoGetClassObject		1
CreateProcessInternalW		2
NtProtectVirtualMemory		33

表 6 CryptoWall ファミリーの API 頻度

どのランサムウェアファミリーでも、違う API シーケンスを持つサンプルが幾つかありる．API シーケンスが違い場合は 2 つあり、一つ目は API シーケンスの内容が全然違う．二つ目は API シーケンスの長さが違い、シーケンス中身の違い所は新しい API 一つ、二つぐらいがあることだ．したがって、本研究で使用した API シーケンスが上記二つ目、長さが違い、中身がほぼ同じの API シーケンスである．

4) 考察

実験で収集したランサムウェアの解析レポートによって、サンプルが同じファミリーに属しても、親プロセスに含まれる API シーケンスが同じではない可能性がある．また、サンプルのプロセスの数が同じであれば、それらのサンプルの親プロセス API シーケンスがほぼ同じであることがわかった．また、ファイル、レジストリなどを操作する API が亜種を分類しに使用できるだと考えている．しかし、同じファミリーに属し、違う親プロセス API シーケンスを持つサンプルの識別仕方を考えなければならない．最後、ランサムウェア検体が足りなかったため、分類実験を実施することができなかったが，解析実験から貰った結果により、同じファミリー属するランサムウェアが同じ親プロセス API シーケンスを持つ傾向があり、違う親プロセス API シーケンスを持つサンプルもあるけれども、親プロセスの API シーケンスが同じであれば、同じファミリーに属することが確定できる．

5. 終わりに

本研究でランサムウェア亜種を分類するために、各ランサムウェアファミリーの親プロセス API シーケンスを抽出し、共通である API 種類を確定した．それらの API シーケンスを使用し、API 間の相関係数を求め、ランサムウェア

亜種を分類できるだと考えている．今後の内容について、十分なランサムウェア検体を収集し、ランサムウェアファミリーを区別する実験を行うことを考えている．

参考文献

[1] 独立行政法人情報処理推進機構, “情報セキュリティ白書 2017”, pp.6, Aug. 2017.
[2] フォーティネットジャパン株式会社, “ランサムウェアの現状と考察”, pp.2, June 2017.
[3] 中国国立インターネット緊急センター(CNCERT/CC), “2017 年中国インターネットサイバーセキュリティレポート”, pp.35, June 2018.
[4] テンセントセキュリティ, “2017 年インターネットセキュリティレポート”, Jan. 2018.
[5] マルウェア情報局, “激増するランサムウェアの傾向の対策”, https://eset-info.canon-its.jp/malware_info/trend/detail/160419.html, April 2016.
[6] I. K. Cho, T. Kim, Y. J. Shim, H. Park, B. Choi, and E. G. Im, “Malware Similarity Analysis using API Sequence Alignments,” J. Internet Serv. Inf. Secur., vol. 4, no. 4, pp.103-114, 2014.
[7] Q. Gong, J. Cao, T. Lu, and D. Li, “Research on detecting Ransomware based on characteristic frequencies,” Application Research of Computers, vol.07, pp.1-2, July. 2018.
[8] P. Natani, D. Vidyarthi, “Malware Detection using API Function Frequency with Ensemble Based Classifier,” International Symposium on Security in Computing and Communication, vol. 377, pp. 378-388, Springer, Berlin, Heidelberg, 2013.
[9] Z. Salehi, M. Ghiasi, and A. Sami, “A Miner for Malware Detection Based on API Function Calls and Their Arguments”, Artificial Intelligence and Signal Processing(AISP), 2012 16th CSI International Symposium on IEEE, pp. 563-568, 2012.
[10] Cuckoo Sandbox, “https://cuckoosandbox.org”.