サイバーレスキュー隊（J-CRAT）技術レポート 2017

インシデント発生時の初動調査の手引き

～WindowsOS 標準ツールで感染を見つける～

独立行政法人情報処理推進機構
セキュリティセンター

目次

1. はじめに

1.1. 本レポートの目的と想定読者

(1) 本レポートの目的

ほとんどの組織が、ネットワークに繋がっているシステムで業務を実施しているが、裏をかえすと日常的にサイバー攻撃の脅威に晒されていることになる。

そうした組織で、

- PC で不審な添付ファイルを開いてしまった
- メールの不審なリンクを誤ってクリックしてしまった
- 不審な USB を不用意に PC に挿入してしまった
- PC の動作や表示に何か違和感がある
- PC から不審な通信が発生している

といった事象（インシデント）が組織内の PC 単位で発生もしくは検知された場合に、どのような対応をとるべきか、この初期段階で、適切な調査とその結果を元に後の対応の判断をできるかが、標的型サイバー攻撃に対抗するためには非常に重要となってくる。

本レポートでは、インシデントの初期段階に、システム管理者が実施すべき「初動調査」について具体的な手順を解説している。なお、この初動調査における各種手法は、J-CRAT [1] のレスキュー活動において初動対応でも活用しているものの一部である。

本レポートは、その初動調査を各組織のシステム管理者や CSIRT 要員が、自ら実施するための手引きである。これによって、万一インシデントが発生した場合にも、組織内で適切かつ迅速な調査が実施され、大きな実被害を回避、抑止、低減できるようになることを期待している。

(2) 想定読者

- 組織のシステム管理をしている部門
- 組織の CSIRT
- サイバー攻撃の被害の調査技術を習得したい技術者

[1] J-CRAT：URL <https://www.ipa.go.jp/security/J-CRAT/index.html>

1.2. インシデント対応における調査の全体像

インシデントが発生すると、システム管理者が把握したいのは以下のようなことである。

1）**攻撃の有無**：これは本当に攻撃だろうか？
2）**感染の有無**：この PC は感染したのだろうか？
3）**被害範囲の把握**：被害はどれくらい進んでしまったのだろうか？
4）**対策の有効性**：対処した対策は有効に機能しているだろうか？

そして、そのインシデントが標的型攻撃の一端だった場合、初期の攻撃が成功すると、その後は組織のネットワークを経由し、他の PC やサーバへと感染拡大する可能性がある。そのため、これらの攻撃や被害範囲の把握には、多くの調査が必要になる。これらの調査の全体像を**表 1.2-1** にまとめた。

表 1.2-1 標的型攻撃の調査の全体像

フェーズ	調査目的	調査	調査対象	調査の観点	関連する調査目的 A	B	C	D	E	F	G
攻撃嫌疑	A. 攻撃メールかどうか	メール調査	PC	・送信元のFQDN/IPアドレスに攻撃インフラの可能性があるかどうか ・添付ファイルがマルウェア等の攻撃であるかどうか ・リンクが正規のものかどうか	○			△			
		不審ファイル調査	PC	・添付ファイルが不審な挙動をしないか ・外部通信が発生する場合、FQDN/IPアドレスに攻撃インフラの可能性があるかどうか	○			△			
感染嫌疑	B. 感染しているかどうか	PC永続化設定調査	PC	・不審な永続化設定がないか（初動調査）		○					
		PC実行痕跡調査	PC	・不審ファイルを実行したかどうか ・ブラウザ等のキャッシュにリンク先が記録されていないか ・不審な通信先への名前解決をおこなっていないか（初動調査）		○	△				
		PC感染頻出箇所調査	PC	・不審ファイルが配置されていないか（初動調査）		○					
		プロキシサーバ調査	SV	・不審ファイルを実行した結果、外部へ通信をおこなっていないか ・リンクをたどって、外部へ通信をおこなっていないか ・FQDN/IPアドレスに攻撃インフラの可能性があるか		○	△		△		
		Firewall調査	NW	・不審ファイルを実行した結果、外部へ通信をおこなっていないか ・リンクをたどって、外部へ通信をおこなっていないか ・FQDN/IPアドレスに攻撃インフラの可能性があるか		○	△		△		
被害の把握	被害の範囲を知りたい C.通信先 D.侵入元 E.被害の範囲 F.情報漏えい	メールサーバ調査	SV	・同時に他メールアドレスあてに受信していないか ・同送信元（サーバ、メールアドレス）から受信記録がないか				△	○		
		プロキシサーバ調査	SV	・同通信先への通信はいつからはじまっているか ・同通信先に組織内の別PCから通信が発生していないか		△	△		○		
		DNSサーバ調査	SV	・同FQDNのクエリがいつからはじまっているか ・同FQDNのクエリが組織内の別PCから発生していないか		△	△		○		
		Firewall調査	NW	・同通信先への通信はいつからはじまっているか ・同通信先に組織内の別PCから通信が発生していないか		△	△		○		
		同一セグメントPC調査	PC	・PC永続化設定・実行痕跡・感染頻出箇所調査をおこなう ・感染PCと同じネットワークアドレスのPCに不審なリモートデスクトップ接続や管理共有への接続などがおこなわれていないか					○		
		組織内PC/サーバ調査	PC SV	・PC永続化設定・実行痕跡・感染頻出箇所調査をおこなう ・感染PCが到達可能なネットワークアドレスのPCに、不審なリモートデスクトップ接続や管理共有への接続などがおこなわれていないか					○		
		ActiveDirecoryサーバ侵害調査	SV	・永続化・実行痕跡・不審ファイル調査をおこなう ・共有フォルダへのアクセス、また、不審なリモートデスクトップ接続や管理共有への接続などがおこなわれていないか ・感染PCからのアクセスについては特に、利用時間、頻度など分析をおこなう					○	△	
		ファイルサーバ侵害調査	SV	・永続化・実行痕跡・不審ファイル調査をおこなう ・共有フォルダへのアクセス、また、不審なリモートデスクトップ接続や管理共有への接続などがおこなわれていないか ・感染PCからのアクセスについては特に、利用時間、頻度など分析をおこなう					○	△	
		初期感染の特定	PC SV NW	・組織内感染がどこからはじまったかを特定する ・メールによる攻撃が疑われる場合は、メールサーバ調査、不審メール調査など ・外部公開サーバからの侵入が疑われる場合は、外部公開サーバ調査、Firewall調査など				○	△		
		感染PC/サーバ詳細調査	PC SV	・状況に応じて感染PCフォレンジックと並行、またはどちらかを実施する ・イベントログ、不審ファイルの各種情報を調査し、感染日、感染方法、感染後の挙動を調査する				△	○	△	
		感染PC/サーバフォレンジック	PC SV	・状況に応じて感染PC詳細調査と並行、またはどちらかを実施する ・イベントログ、不審ファイル、ファイルシステム、レジストリ等の各種情報を調査し、感染日、感染方法、感染後の挙動を調査する				△	○	△	
対策の有効性	G.実施した対策が有効なことを知りたい	プロキシサーバ調査	SV	・同通信先への通信が発生していないか ・同通信先に組織内の別PCから通信が発生していないか					△		○
		DNSサーバ調査	SV	・同FQDNのクエリが発生していないか ・同FQDNのクエリが組織内の別PCから発生していないか					△		○
		Firewall調査	NW	・同通信先への通信が発生していないか ・同通信先に組織内の別PCから通信が発生していないか					△		○

○：直接的な効果　△：間接的な効果

この表は、インシデント検知時に、今実施する調査が何で、次に着手すべき調査が何なのかを、「調査の目的」から、「調査対象」と「調査の観点（どのような調査をするか)」を得るための早見表である。

例えば、「感染しているかどうか」を調べるためには、PC で「永続化設定」「実行痕跡」「感染頻出箇所」（本レポートで初動調査として解説）に関する調査を行い、あわせて、「プロキシサーバ」「Firewall」にて、感染嫌疑 PC から外部へ不審な通信が出ていないかを調査すべき、であることを示している。

また、調査全体から見た場合には、同じ調査でも、フェーズによって「調査目的」が変化する。例えば、感染嫌疑時の「プロキシサーバ」調査（不審な通信先がないか）と、感染後の被害の把握時の「プロキシサーバ」調査（その不審先へいつから、どこから通信しているか）は、調査目的が異なることを示している。

そして、それぞれの調査は主たる単一の目的だけでなく間接的な意味もあることを、「関連する調査目的」として、直接的な効果があるものを○、間接的な効果があるものを△で示した。例えば、被害の把握時において、初期感染を特定することは、侵入元を探すことではあるが、被害の範囲を調査する意味もあるということを示している。

この表を「システム管理者の把握したい内容」を４つのフェーズとし、それらの段階別に各調査を表現すると、図 1.2-1 になる。

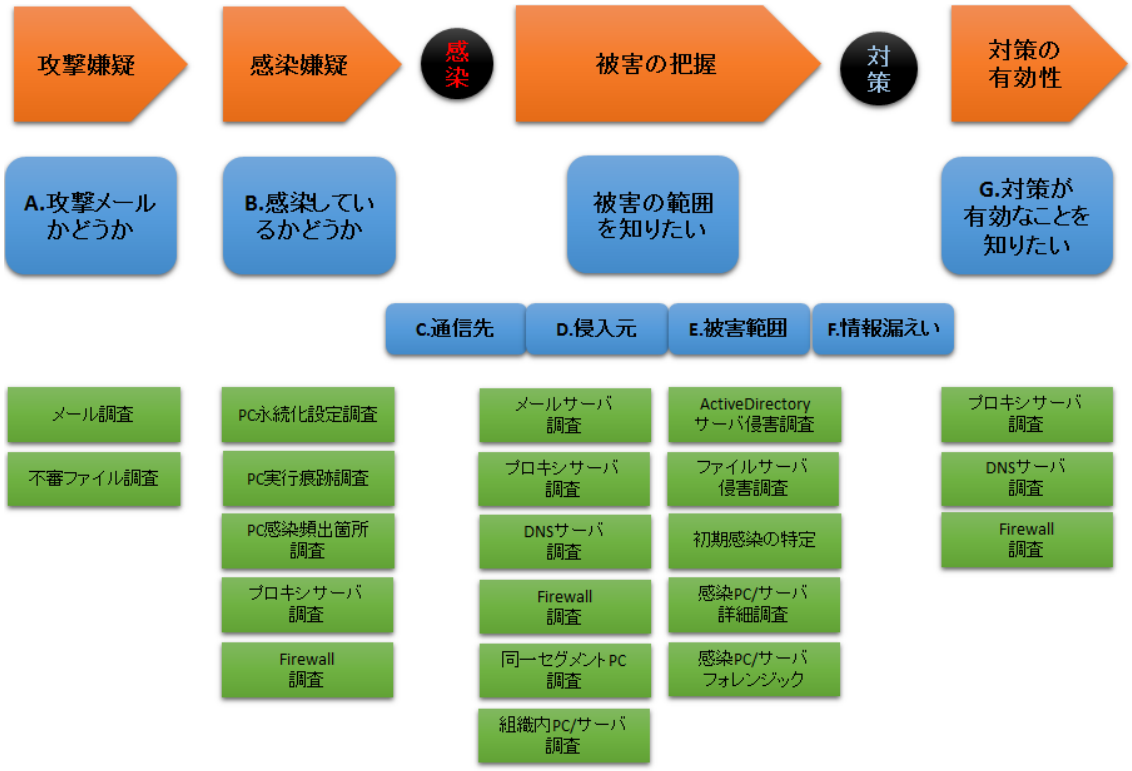


図 1.2-1 フェーズと調査

「感染」後には、多くの調査が必要になることを示している。
そして、これらの調査を「調査の対象」別に表現すると、図 1.2-2 になる。

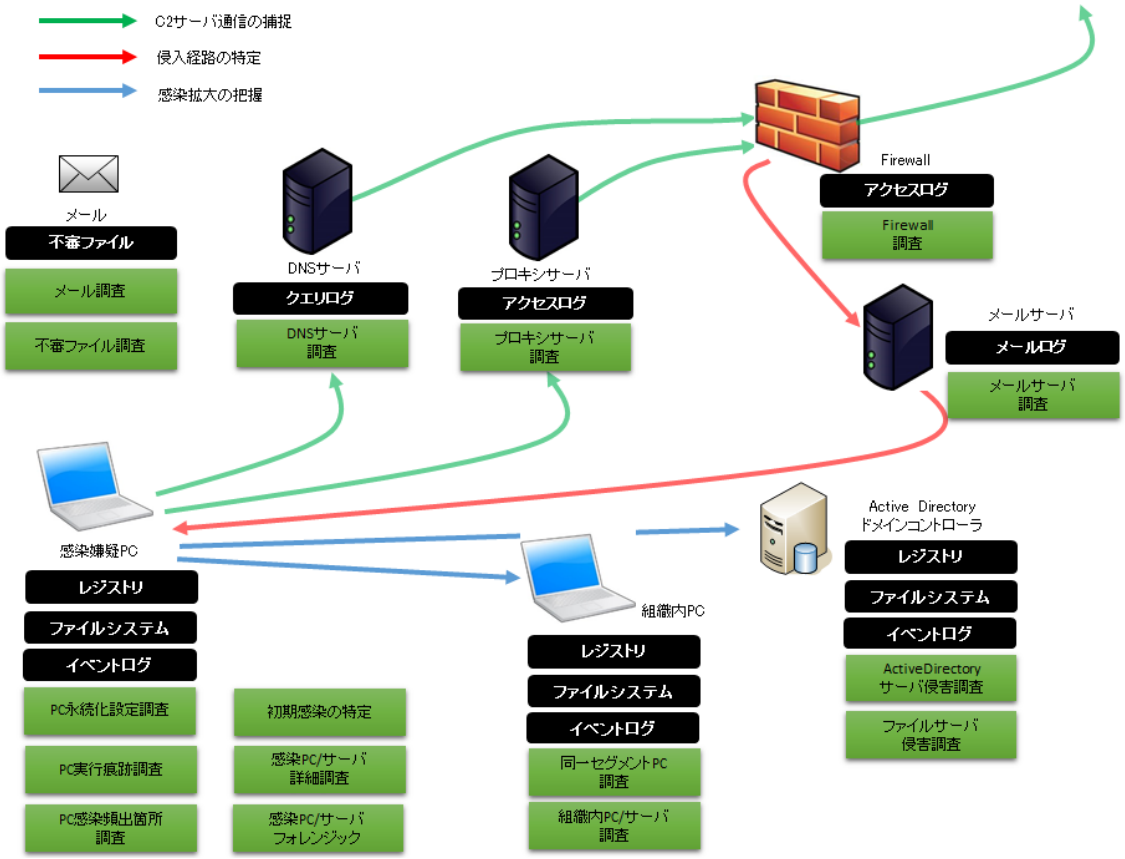


図 1.2-2 調査の対象

調査の対象は、PC、サーバ、ネットワークに分けられる。組織全体から見たときに、サーバやネットワークといった箇所の調査は効果的、効率的であるが、初期感染や感染拡大には PC を経由して行われる標的型攻撃においては、PC の調査が不可欠であることを示している。

1.3. 本レポートでの対象範囲と前提環境

(1) 本レポートでの対象範囲

インシデント検知当初にどこまで調査を行うかは、多くのシステム管理者の悩みどころだ。そこで、マルウェア感染の有無や被害を、ある程度の範囲を把握することを目的とし

た PC に対して最初に行うべき調査を、「初動調査」として提案する。それは、下記の３つの調査からなる。

- **PC 永続化設定調査**
- **PC 実行痕跡調査**
- **PC 感染頻出箇所調査**

調査の対象は、標的型攻撃に用いられるマルウェアへの感染の嫌疑がかかる PC であり、これらのマルウェアがよく使う手法、よく狙う箇所を中心に情報収集を行う。そして、それらの中に不審点がないかを評価する。

この初動調査を、インシデントの発生や検知したタイミングで行うことで、図 1.3-1 に示すように、その後の各種調査やベンダー追加調査へ進むかどうかの判定ができる。そして、この調査の結果を受けて、早急な対応や、あるいは外部機関やベンダーへの適切な支援を依頼することが可能となる。

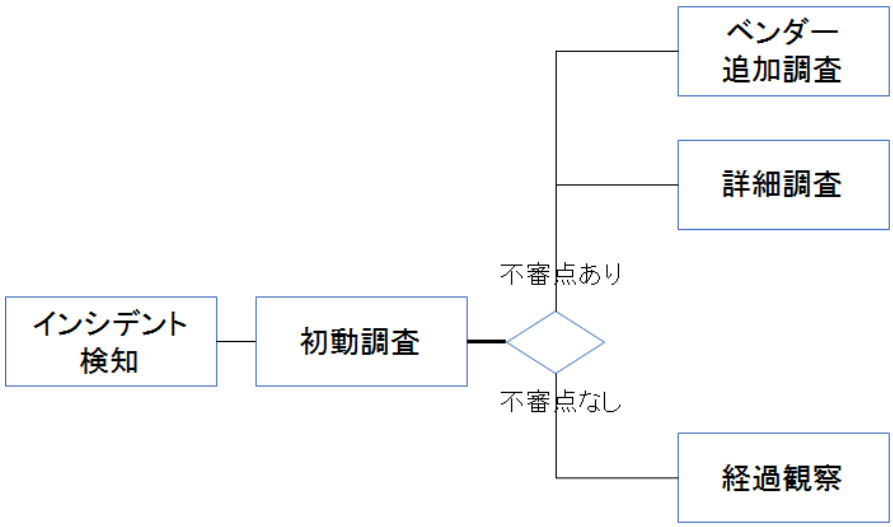


図 1.3-1 初動調査フロー

初動調査とはいっても、攻撃者の活動は感染 PC を起点に行われることから、調査対象の PC でひとつの標的型マルウェアの感染が発見されると、そのマルウェア感染以外の活動痕跡が発見されることも多い。標的型攻撃の調査全体から見ても、そこから得られる情報は、感染の有無（感染嫌疑）の確認に留まらず、幅広い調査目的への情報ともなる。

そして、攻撃者が使うツールや手法は、被害組織内での攻撃活動においては多くの共通点が見られる。初動調査で得られた情報は、組織内の他 PC での調査において、有効な情報源になるものである。

(2)　前提環境

本レポートでは、情報収集する対象の PC は WindowsOS を対象としている。紹介するコマンド実行を試したバージョンは下記の通りである。

- Windows 10 Professional
- Windows 7 Professional

情報収集した結果の評価は、情報収集の対象 PC とは異なる PC で実施することを想定している。その理由は以下の事態をさけるためである。

- 不審ファイルの意図しない実行
- 誤操作によるファイルや設定等の消失
- ログの追加記録による過去ログの消失

また、紹介するオープンソースのツールは以下の環境が必要である。事前に準備していただきたい。

- Python2.7

不審点の洗い出しの作業は、具体的には情報収集した結果である「大量のテキストデータ」から、攻撃痕跡に紐付きそうなキーワードを抽出する文字列検索である。本レポートでは、Window7 以降の OS に装備されているコマンド[2]を中心に記載している。

(3)　本レポートの構成

本レポートは、下記の章構成となっている。各読者のニーズにあわせて、必要な章を参照いただきたい。

章	内容
１章　はじめに	標的型攻撃における調査の全体像、初動調査の定義、初動調査の位置づけ、本レポートの特徴など。
２章　基礎知識	初動調査の前提となる標的型攻撃の特性を解説。
３章　情報収集	情報収集するためのコマンドを実行例とともに紹介。

[2]可能であれば、評価環境は LinuxOS で構築することをお奨めする。各種の強力な文字列検索機能が標準装備されていることと、何より WindowsOS のマルウェアを誤動作させにくいという利点があるからだ。

４章　評価	評価するポイントを実際の感染例とともに紹介。
５章　おわりに	本レポートのまとめ、情報提供のお願い。

(4)　表の色分けと引用について

本レポートでは２章以降、コマンド構文や実行例とあわせて実際の感染例や痕跡例も多く記載している。それらの表は色分けをしており、実行例を青色、感染例は灰色としている。

コマンド構文・実行例
内容

感染例・痕跡例
内容

また、引用しているコマンドや表、感染例の一部は、過去に J-CRAT が発行した各種資料と重複している。感染例・痕跡例の中に含まれていた組織名や IP アドレス等の情報は架空のものに変更している。

(5)　注意点

本レポートで紹介している内容はあくまで「調査の一例」であり、「PC へのセキュリティ対策」ではない。OS やアプリケーションの最新化、ActiveDirectory や PC 単位の適切なセキュリティ設計、適切なウイルス対策ソフトの運用（必要に応じたエンドポイントセキュリティの実装)、資産管理ツールの活用など、本来あるべきセキュリティ対策の実施は必要不可欠である。

また、本レポートで紹介する内容には、感染嫌疑 PC でコマンドを実行して行う情報収集や、不審点を抽出するための公開情報の調査を含んでいる。前者の情報収集作業は、コマンド実行するには、OS を動作させた状態でなければできない。そのため、ログやファイルシステム等の状況は変化する可能性があり、証拠保全の観点からは好ましくない場合がある。感染状況などにより、ディスク保全を行った後に実施するなど、都度判断されたい。後者の公開情報の調査は「攻撃者から見れば」組織が攻撃に気付いたことを知らせてしまう可能性もある。インシデント対応の対応方針については諸説あり、組織によって異なる考え方がある。これらの注意点を理解した上で活用していただきたい。

2. 基礎知識

2.1. マルウェア感染の特性

まず、標的型攻撃で利用されるマルウェアに感染したPCに見られる特性を解説する。

- 永続化
- 偽装
- 外部通信

2.1.1. 永続化

攻撃者は多くの場合、RAT（Remote Access Tool）を利用する。外部ネットワークからリモートアクセスし、そのPCで様々な操作を行うことで情報収集、情報窃取し、ネットワーク経由で横移動していく。しかし、攻撃者は、感染直後にすべての攻撃活動を完了させるとは限らない。そこで、感染させた状態を維持しておくため、RATやダウンローダーといったツールをOS起動時に自動的に起動するようPCの設定を変更する。これを永続化という。標的型攻撃のマルウェアに感染していたPCを調査すると、殆どの場合でなんらかの永続化設定が見られる。

WindowsOSではおおよそ以下の設定箇所がある。

- 自動起動レジストリ
  - スタートアップ起動プログラム
  - サービス起動プログラム
- スタートアップフォルダ
- ログオンスクリプト
- タスクスケジューラ

このような設定箇所に、意図しないプログラムが起動する設定になっていないか、覚えのない設定がいつのまにか追加されていないかをチェックすることで不審点を調べることができる。

2.1.2. 偽装

永続化されたマルウェアは多くの場合、フォルダ名やファイル名、サービス名などを偽装する。利用者から発見されにくいようにするための工夫である。その手法は以下のようなものである。

- 著名なアプリケーションと同じ、または似せた名称をフォルダ名、ファイル名、サービス名に使用する。
- Windows の正規ファイルと同じ、または似せた名称をファイル名に使用する。

また、マルウェアの起動時に偽装を行うケースもある。例えば、以下のような手法がある。

- Windows 標準の正規プログラムを利用する。
- 正規アプリケーションの一部のプログラムを利用する。

このような方法だと、起動中のプログラム名は正規ファイルとなるため、マルウェアが起動していても利用者は気付きにくい。

偽装を見破るのは難しいが、自身がインストールした記憶がないアプリケーションがないか、実行ファイルが配置された場所に不自然な点がないか、といった観点で不審点を洗い出していく。

2.1.3. 外部通信

先に説明した通り、攻撃者は外部から攻撃対象 PC を操作しようとする。攻撃用に送り込んだマルウェアから、攻撃者が準備した Command & Control サーバ（以降 C2 サーバ）へ通信をさせることで、そのネットワーク接続を確立させる。そして多くのマルウェアはその接続時に使うポートは、組織で一般的に許可されるポート、例えば HTTP や HTTPS、DNS のようなものを利用することで、正規通信に紛れ込ませるという手法をとる。このため、PC が行った外部通信の中に、C2 サーバとの通信が紛れ込んでいないかを調べることが必要となる。

本レポートでは、PC を対象とした調査を想定しているため、PC の DNS リゾルバが残している「DNS キャッシュ」を評価の対象とする。

ただし、マルウェアによっては、IP アドレスで直接通信先が指定されている場合や、インターネット接続がない場合には通信をしないこともあるため、必ずしも DNS キャッシュに記録が残るわけではない。

情報収集時に、もしネットワークを抜線していない状態で実行できる場合は、調査時のネットワークの状態を出力する「ネットワーク接続情報」もあわせて取得する。

これらのログや出力結果に、意図しない通信先、不審な通信先がないかを確認することで、C2 サーバとの通信を発見できることがある。

不審な通信先が発見されたら、外部通信記が残っている可能性のあるプロキシサーバや DNS サーバ、Firewall といった機器上のログ調査もあわせて行うことで、攻撃痕跡や感染の範囲など、より具体的に把握することにつながる。

2.2. 攻撃痕跡を判断するために

次は、攻撃の痕跡が残りやすい箇所を解説する。

2.2.1. マルウェアが配置されやすい場所(感染頻出箇所)

マルウェアが配置されやすいフォルダは存在する。攻撃者が利用しやすいフォルダと言い換えてもよい。特に管理者権限がないユーザーが権限を有し、環境変数でアクセスしやすい箇所がそれにあたる。

表 2.2-1 感染頻出箇所

環境変数等	実フォルダ例
%TEMP%	C:¥Users¥%USERNAME%¥AppData¥Local¥Temp
%PROGRAMDATA% %ALLUSERSPROFILE%	C:¥ProgramData
%APPDATA%	C:¥Users¥%USERNAME%¥AppData¥Roaming
%LOCALAPPDATA%	C:¥Users¥%USERNAME%¥AppData¥Local
%PUBLIC%	C:¥Users¥Public

マルウェアに感染した端末を調査すると、これらのフォルダにマルウェアやツールが配置されることが多い。これらのフォルダの共通点は、「一般利用者権限、またはログインユーザー」に対して多くの権限が与えられていることだ。そのため、この場所を利用すれば、標的型攻撃の主な感染源となっているメールによる攻撃でも成功する可能性が高い。マルウェアの動作を確認すると、そのファイルを開いたときに%TEMP%にマルウェアが一旦生成され、%PROGRAMDATA%以下に偽装したフォルダにコピーし、そのファイルを永続化設定に定義するといった例も多い。

もちろん、マルウェアの実行ファイルが、感染頻出箇所ではない場所にあることもある。感染時の利用者に与えられた権限や感染 PC の設定にも関連するが、攻撃者の活動により権限昇格や管理者権限が奪取されれば、場所に限定されない配置が可能となる。しかし、初期感染時にこれらの感染頻出箇所を狙うことは、攻撃者にとって効率がよいと考えられる。また、複数のフォルダを用途によって使い分けていると思われる例もあり、多くの痕跡が残されることからも、インシデント検知時にはこれらのフォルダを調べることを推奨する。

参考情報）ドライブ直下のフォルダ

攻撃ツール等の置き場所として、デフォルトにはない C:¥直下のフォルダが利用されていることがある。例えば、C:¥Temp や C:¥Intel などだ。これは、C:¥Windows 等には、細かなアクセス制限がかかっているが、C:¥ドライブ直下は、ログインユーザーに書込み権限が与えられているため、フォルダの作成等が可能であり、攻撃者も利用しやすいと言える。不審点や感染が発見された場合は、このようなフォルダがないか、またそのフォルダに配置されたファイルの確認を推奨する。

2.2.2. 実行痕跡

マルウェア、またはマルウェアを内包したファイルを実行したかどうかを調査することは、重要な情報になる。

実行痕跡は、単に動作したアプリケーション上の動作記録としてのログだけでなく、実行を示唆するものであればよいため、様々な箇所の記録が活用できる。以下に例をあげる。

- イベントログ*
- アプリケーション固有ログ*
- アプリケーション実行記録
  - PreFetch Files
  - 最近使ったファイル
  - 最近使った Office ドキュメント
  - AppComPatCache
  - RecentFiles
  - UserAssist**
  - RunMRU
  - TypedURL

- ウイルス対策ソフトの検知ログ*
- フォルダ、ファイルのタイムスタンプ

これらのうち、本レポートで取り扱うのは、*および**以外のものとする。*印のものは、各種情報を記録することを目的にしたログであり、それらの調査は各アプリケーションの製品情報が必要であるため、対象外としている。初動調査の結果によって、より詳細の調査に進む場合は、これらのログ調査を行ってほしい。また、**印のものについては、本レポート公開時に適切な解析ツールが発見できなかったため、記載していない。

アプリケーション実行記録については、今回は「実行したアプリケーションを把握しやすい」と思われる代表的なものをあげている。

表 2.2-2 代表的な実行痕跡

実行痕跡	内容と方法
Prefetch Files	アプリケーション起動時の各種情報をファイルとして保持。C:¥Windows¥Prefetch フォルダにファイル名-フルパスハッシュ値.pf として作成される。ファイル名取得と更新日による実行判断と実行日付が推測できる。さらに、pf ファイルそのものを解析すると、起動時の読み込みファイルや実行回数等を確認できる。128 個保存される。
最近使ったファイル	エクスプローラー経由で「使った」ファイル名から C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Windows¥Recent フォルダにファイル名.lnk ファイルが作成される。開封判断に利用できる。
最近使った Office ドキュメント	Office ドキュメントを「使った」ファイル名から、C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Office¥Recent フォルダにファイル名.lnk ファイルが生成される。
AppCompatCache	アプリケーション実行時のキャッシュ情報としてレジストリに保持している。フルパスを含む実行ファイル名、最終更新日、サイズ、ファイルの実行可否が記録される。1024 個保存される。
UserAssist	エクスプローラー経由で実行したプログラム情報をレジストリに保持している。
RunMRU	「ファイル名を指定して実行」で実行したプログラム情報をレジストリに保持している。
TypedURLs	InternetExplorer でアクセスした直近の URL が保持されている。25 個から 50 個が保存される。

必ずしも、同一の情報が記録されるわけではないが、アプリケーションが実行された痕跡は、このような箇所に残されている可能性がある。これらは、バイナリでの出力やエンコードされているものもあるため、それらの評価は、解析ツールで、可読性のあるテキストに出力してから行う。

フォルダやファイルのタイムスタンプは、マルウェアの実行時に意図的に操作されることも多いため、考慮にいれる必要がある。しかし、例えばエラー時に生成されるファイルのように、OS の機能によって間接的に生成されるものは、その最終更新時間などが感染時間を推測する要素になる。

2.2.3. 攻撃によく使われるツール

標的型マルウェアに感染した PC では、複数のマルウェア、ツールが発見されることがある。権限奪取や設定変更に使われるツールの名前を知っておこう。ここでは、比較的よく利用されるものをあげる。そして、これらのうちの多くは正規ツールである。これらの実行痕跡があり、意図して実行したものでなければ、何らかの攻撃である可能性も考えられる。

表 2.2-3 攻撃によく使われるツール

ファイル名	機能など
PowerShell.exe	WindowsOS 標準。高機能なスクリプト言語。近年特に攻撃での利用が多い。
wmic.exe	WindowsOS 標準。WindowsOS の管理基盤アーキテクチャ WMI（Windows Management Instrumentation）にアクセスするためのコマンドラインツール。
at.exe	WindowsOS 標準。スケジュール作成コマンド。 リモートコンピュータでのプログラム実行にも利用できる。
schtasks.exe	WindowsOS 標準。タスクスケジューラのタスクを操作するコマンド。リモートコンピュータでのプログラム実行にも利用できる。
Psexec.exe	Windows Sysinternals に含まれるリモートシステムでのプログラム実行ツール。組織内で感染を拡大する「横移動」によく使われる。
wce.exe	著名なハッキングツール。Windows Credential Editor の略で、ログオン中のパスワードを出力する。
Mimikatz.exe	著名なハッキングツール。メモリ上に保持されているアカウントの認証情報にアクセスし、各種クレデンシャル情報を奪

	取する。管理者権限の奪取によく使われる。様々な形態があり、PowerShell 版も存在する。
gsedump.exe	著名なハッキングツール。パスワードハッシュを出力する。

2.3. 情報収集作業の事前知識

情報収集にあたり、知っておくべき WindowsOS におけるレジストリとユーザープロファイルについて解説する。

情報収集は、主に「レジストリ」「特定のコマンド実行結果」「ファイル一覧」を取得する。これらのうち、「レジストリ」と「ファイル一覧」には、システム全体に紐付くもの、ユーザーに紐付くものがあるため、収集する際に、それがどちらであるかを知っておく必要がある。

(1) レジストリ

レジストリは、WindowsOS の管理情報の一種で、ツリー構造で管理されており、ルートキーから各サブキーに枝分かれして構成要素毎に値とデータを保持している。

本レポートで取り扱うレジストリのルートキーは主に以下の２つである。

表 2.3-1 主なルートキー

ルートキー	省略形	意味
HKEY_LOCAL_MACHINE	HKLM	システム全体にかかわる各種構成要素
HKEY_CURRENT_USER	HKCU	現在ログオンしているユーザーの各種構成要素

次章で、具体的な収集コマンドを記載しているが、レジストリに関しては、ルートキーの指定を”HKLM”や”HKCU”という省略形で記載している。

注意が必要なのは、”HKCU”は、現在ログオンしているユーザーのレジストリとなるため、インシデント検知時のログオンユーザーと情報収集時のログオンユーザーが異なると情報収集する内容も異なることだ。初期感染の場合、ログオンユーザーがアクセスしやすい箇所に感染することが多いため（2.2.1 参照）、情報収集時もそのログオンユーザーで行うことを推奨する。

また、複数のユーザーで PC を共有している場合も同じ理由で、感染頻出箇所が複数あるということになるので、ログオンし直して情報収集することも検討してほしい。

(2)　ユーザープロファイル

レジストリと同様に、ユーザー毎に管理されているものとして、「ユーザープロファイル」がある。ユーザー毎に作成される「フォルダ」に、まとめて各種ファイルを保存するという仕組みである。よって、複数のユーザー[3]が PC を利用した場合は、このユーザープロファイル用のフォルダがそれぞれ作成されている。

ユーザープロファイルは wmic コマンドで出力することが可能だ。デフォルトでは C:¥Users フォルダ以下に生成されたサブフォルダに保存されている。

コマンド例
wmic path win32_userprofile

実行例（抜粋）			
Loaded	LocalPath	RefCount	RoamingConfigured
FALSE	C:¥Users¥user02	0	FALSE
TRUE	C:¥Users¥user01	3	FALSE

実際の C:¥Users には以下のフォルダが生成されている。

ファイル例			
C:¥Users>dir /ad			
2018/01/22	09:11	<DIR>	.
2018/01/22	09:11	<DIR>	..
2018/01/22	10:56	<DIR>	user01
2018/01/22	10:56	<DIR>	user02
2017/12/22	10:19	<DIR>	Public

デフォルトで作成されているフォルダである“Public”以外の“user01”と“user02”が、この PC に保存されているユーザープロファイルである。

本レポートでは、ファイル一覧を取得する際、%USERNAME%という変数で指定している箇所があるが、取得時のログオンユーザー名がその変数に代入されているため、ログオン中のユーザーのユーザープロファイル配下を指すことになっている。よって、レジストリと同じく、調査対象となるユーザーアカウントでのログオン時に情報収集することを推奨する。

[3] ActiveDirectory の設計等によっては、PC ローカルにユーザープロファイルを置かない場合もある。

ただし、レジストリと違って、このプロファイルはファイルシステム上のフォルダであるため、読み込み権限がある場合は、別ユーザーでログオンしていても参照可能である。その特性を利用して、別ユーザーのユーザープロファイル配下のファイル一覧を収集することも可能だ。

2.4. 共通する手法とツール

情報収集や評価の際に共通する手法とツールについて解説する。「リダイレクト」「findstr」「パイプライン」の３つだ。

(1) リダイレクト

情報収集の際は、各種コマンドを実行することを行うが、評価のためには、それらの実行結果をファイルとして保存しておく必要がある。本レポートで紹介するコマンドやツールで、実行結果をファイルとして保存する機能がないものについては補完する必要がある。ここでは「リダイレクト」という方法を紹介する。

コマンドの実行結果を別のデバイスへ変更することを「リダイレクト」という。このデバイスを任意のファイルとして保存する。リダイレクトは、下記のようにコマンドの後ろに「>」記号と任意のファイル名を指定する。

コマンド構文
コマンド > 任意のファイル名

実行例
C:¥Windows¥system32>reg query "HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run" /s >c:¥tmp¥hklmrun.txt

この実行例では、レジストリ検索コマンドの結果を、c:¥tmp フォルダに“hklmrun.txt”というファイルとして保存している。なお、リダイレクトした場合、出力結果は画面に表示されない。

(2) findstr

評価作業の実際は、情報収集した各種の出力結果である（主に）テキストデータから、様々な検索対象キーワードから文字列検索コマンドを使って抽出する作業である。ここでは、WindowsOS に標準搭載されている「findstr」を紹介する。

コマンド例
findstr “検索対象文字列” 検索対象ファイル

コマンド構文（抜粋）

```
FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P]
        [/F:ファイル] [/C:文字列] [/G:ファイル] [/D:ディレクトリ一覧]
        [/A:色属性] [/OFF[LINE]] 文字列 [[ドライブ:][パス]ファイル名[ ...]]

  /B            行の先頭にあるパターンを検索します。
  /E            行の末尾にあるパターンを検索します。
  /L            検索文字列をリテラルとして使用します。
  /R            検索文字列を正規表現として使用します。
  /S            現在のディレクトリとすべてのサブディレクトリから一致する
                ファイルを検索します。
  /I            検索するときに大文字と小文字を区別しません。
  /X            完全に一致する行を出力します。
  /V            一致しない行のみを出力します。
  /N            一致する各行の前に行番号を出力します。
  /M            ファイルに一致する行があるときにファイル名のみを出力します。
  /O            一致する各行の前に文字オフセットを出力します。
  /P            印刷不可能な文字を含むファイルをスキップします。
  /OFF[LINE]    オフライン属性が設定されたファイルをスキップしません。
  /A:属性       2 桁の 16 進数で色属性を指定します。"color /?" を参照してくだ
                さい。
  /F:ファイル   指定したファイルからファイル一覧を読み取ります (/ を指定する
                とコンソール)。
  /C:文字列     指定された文字列をリテラル検索文字列として使用します。
  /G:ファイル   指定されたファイルから検索文字列を取得します (/ を指定する
                とコンソール)。
  /D:ディレクトリ
                セミコロンで区切られた検索されるディレクトリ文字列テキストの
                一覧を検索します。
  [ドライブ:][パス]ファイル名
                検索するファイルを指定します。
```

複数の検索対象キーワード、検索対象の複数指定、検索文字列のファイル化、正規表現など、多くのオプションに対応している。これらを組み合わせることで、感染嫌疑の PC 台数が多い場合や組織内での同件確認の際に、キーワードをファイル化して共有など、様々な対応が可能になる。

(3) パイプライン

文字列検索を行う際には、あるキーワードで抽出した結果から、さらに別のキーワードを使って抽出するといった作業がよく行われる。そのような場合、よく利用される「パイプライン」を紹介する。

コマンド例
コマンド１|コマンド２

連結させたいコマンドを「|」記号でつなぐことで、パイプラインを実行できる。コマンド１の結果をコマンド２に処理させることができる。

例えば、“C:¥Windows¥System32”のファイル一覧から、“dll”の拡張子を持つファイルを抽出したいとする。これをパイプラインで連結すると以下のようになる。

コマンド例
dir C:¥Windows¥System32 | findstr ″¥.dll″

コマンド実行例
C:¥>dir c:¥windows¥system32 |findstr ″¥.dll″
2015/07/10 17:24 26,112 aadauthhelper.dll
2015/07/10 17:24 224,768 aadcloudap.dll
2015/07/10 17:24 491,520 aadtb.dll
2015/07/10 17:25 126,464 AboveLockAppHost.dll
2015/07/10 17:24 3,789,312 accessibilitycpl.dll
2015/07/10 17:24 161,280 accountaccessor.dll
2015/07/10 17:25 12,800 AccountsControlInternal.dll
2015/07/10 17:24 306,176 AccountsRt.dll

この章では、情報収集と評価のための基礎知識を記載した。次章では、具体的な収集項目とコマンドの解説を行う。

3. 情報収集

前章で述べた「標的型攻撃でよく見られる特性」をもとに、それらの痕跡が残りやすい箇所を中心に、WindowsOS 標準コマンドを使って、情報収集を行う。収集すべき項目と収集するコマンドを、以下の４つに分けて解説する。

- 永続化設定
- 外部通信
- 実行痕跡
- 感染頻出箇所

収集対象は、レジストリやキャッシュ、特定ディレクトリのファイル一覧などであり、それぞれに適したコマンドで収集する。

3.1. 永続化設定の取得

永続化設定のうち、下記の４つがマルウェア感染によく使われる。

- スタートアップ起動プログラム
- サービス起動プログラム
- スタートアップフォルダ
- タスクスケジューラ

本レポートでは以下の方法で収集する。

表 3.1-1 永続化設定と収集方法

永続化設定	収集方法
スタートアップ起動プログラム	レジストリ操作ツール
サービス起動プログラム	レジストリ操作ツール
スタートアップフォルダ	レジストリ操作ツール リンクファイル情報の収集 PowerShell によるリンクファイル内容の収集
タスクスケジューラ	タスク管理ツール

3.1.1. スタートアップ起動プログラム

システムのスタートアップ時に起動されるプログラムをレジストリから取得する。ここでは、HKEY_LOCAL_MACHINE と HKEY_CURRENT_USER、そして HKU の起動時に実行されている設定をあげる。[4]

主なスタートアップ起動プログラム設定レジストリ
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Run
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce
HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon : Userinit
HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon : Shell
HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon¥Notify
HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows : Appinit_Dlls
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Run
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run
HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows : Load
HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows : Run
HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon : Shell
HKU¥.DEFAULT¥Software¥Microsoft¥Windows¥CurrentVersion¥Run

これらの設定は、レジストリ操作ツール“reg.exe”で収集できる。

コマンド構文
reg query レジストリ名 オプション

[4] [参考]Windows7 の自動起動
URL<https://technet.microsoft.com/ja-jp/library/ee851671.aspx>

適切な値が出力できるよう「/s（サブキーと値を再帰的に行う)」と一部のものは文字列検索（“”で囲んだ部分）をあわせたオプションを加えて実行する。[5]

コマンド例
reg query "HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run" /s
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Userinit"
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Shell"
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon¥Notify" /s
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "AppInit_DLLs"
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Run" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run" /s
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "Load"
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "Run"
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Shell"
reg query "HKU¥.DEFAULT¥Software¥Microsoft¥Windows¥CurrentVersion¥Run" /s

結果は画面に出力される。設定されていない箇所は空白が出力され、該当するレジストリーキーが存在しない場合は、エラーが出力される。

実行例
HKEY_LOCAL_MACHINE¥software¥microsoft¥windows¥currentVersion¥run Adobe ARM REG_SZ "C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥AdobeARM.exe"

[5] 赤文字部分がオプション。

この実行例では、“C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥AdobeARM.exe”が“Adobe ARM”として登録されていることがわかる。

3.1.2. サービス起動プログラム

サービス起動プログラムもレジストリから収集できる。下記のレジストリに設定されている。

サービス起動プログラム設定レジストリ
HKLM¥SYSTEM¥currentControlSet¥services

コマンド例
reg query HKLM¥SYSTEM¥currentControlSet¥services /s

実行すると、大量の結果が出力される。リダイレクトして、テキストファイルとして保存する。一例を下記にあげる。

実行例
HKEY_LOCAL_MACHINE¥system¥CurrentControlSet¥Services¥AdobeARMservice Type REG_DWORD 0x10 Start REG_DWORD 0x2 ErrorControl REG_DWORD 0x0 ImagePath REG_EXPAND_SZ "C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥armsvc.exe" DisplayName REG_SZ Adobe Acrobat Update Service ObjectName REG_SZ LocalSystem Description REG_SZ Adobe Acrobat Updater はアドビソフトウェアを最新の状態に保ちます。

“ImagePath”の箇所に、サービスとして実行されるファイルが指定されている。この実行例では、“C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥armsvc.exe”が“Adobe Acrobat Update Service”サービスとして指定されているのが見てとれる。

3.1.3. スタートアップフォルダ

このスタートアップフォルダへの配置も、マルウェア配置によく使われる。直接、実行ファイルが配置されるケースや、リンクファイルを偽装するケースなどがある。よって、以下の３つの手順で確認する。

１）スタートアップフォルダの位置の確認
２）指定されているフォルダのファイル一覧確認
３）リンクファイルの内容確認

特に、昨今このリンクファイルに PowerShell のスクリプトを直接記述する攻撃が散見されるようになったため、リンクファイルの内容確認はしておきたい。

(1) スタートアップフォルダの位置の確認

スタートアップフォルダの位置は、下記レジストリに設定されている。

設定レジストリ
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders:Startup
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders:Startup
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders:Startup
HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders:Startup

このレジストリの値“Startup”を照会するよう指定して収集する。

コマンド例
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders" /v "Startup"
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders" /v "Startup"
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders" /v "Startup"
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders" /v "Startup"

実行例
HKEY_CURRENT_USER¥software¥microsoft¥windows¥currentversion¥explorer¥shell folders
Startup REG_SZ C:¥Users¥user01¥AppData¥Roaming¥Microsoft¥Windows¥Start Menu¥Programs¥Startup
HKEY_CURRENT_USER¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders

```
    Startup    REG_EXPAND_SZ    %USERPROFILE%¥AppData¥Roaming¥Microsoft¥Windows¥Start
Menu¥Programs¥Startup
```

この実行例では、“%USERNAME%¥AppData¥Roaming¥Microsoft¥Windows¥Start Menu¥Programs¥Startup”の位置であることがわかる。

(2) フォルダの位置の確認

つぎに、設定されていたフォルダのファイル一覧を取得する。「/q オプション」をつけて、所有者をあわせて出力する。

実行例

```
dir /q /r /s "C:¥Users¥user01¥AppData¥Roaming¥Microsoft¥Windows¥Start
Menu¥Programs¥Startup"
ドライブ C のボリューム ラベルがありません。
 ボリューム シリアル番号は BA8B-6224 です

 c:¥users¥user01¥AppData¥Roaming¥Microsoft¥Windows¥Start Menu¥Programs¥Startup のディレクトリ

2018/03/22  18:51    <DIR>          PC01¥user01 .
2018/03/22  18:51    <DIR>          PC01¥user01 ..
2018/03/22  18:47               908 BUILTIN¥Administrators startup - ショートカット.lnk
```

この実行例では、“startup”のリンクファイルが配置されている。

(3) リンクファイルの確認

スタートアップフォルダにあるリンクファイルの記述内容を出力する方法を紹介する。専用のコマンドは存在しないが、PowerShell のフィルター関数を使えば出力可能だ。以下の例では、%AppData%と%ProgramData%に存在するものとして記載している。

コマンド例
powershell.exe -Command "& {filter Get-ShortCut{$shell=new-object -comobject△ WScript.Shell;return△ $shell.CreateShortcut($_);}$path=$env:APPDATA+¥"¥Microsoft¥Windows¥Start△ Menu¥Programs¥Startup¥";dir $path -r -include ¥"*.lnk¥" |△ Get-ShortCut;$path=$env:PROGRAMDATA+¥"¥Microsoft¥Windows¥Start△ Menu¥Programs¥Startup¥";dir $path -r -include ¥"*.lnk¥"	Get-ShortCut;}"

このコマンド例の「$path=$env:フォルダ;」の箇所を、レジストリ設定の値にあわせることで、該当フォルダのリンクファイルの内容を確認することができる。なお、このコマンドは１行で実行する。各行末には半角空白の代替文字として△を記載した。

実行すると、リンクファイルの情報が下記のように出力される。

実行例（Appdata 抜粋）
FullName : C:¥Users¥user01¥AppData¥Roaming¥Microsoft¥Windows¥Start Menu¥Programs¥Startup¥startup - ショートカット.lnk Arguments : Description : Hotkey : IconLocation : ,0 RelativePath : TargetPath : C:¥tmp¥startup.bat WindowStyle : 1 WorkingDirectory : C:¥tmp

“FullName”がリンクファイル、“TargetPath”が、その実行ファイルを指定している。この出力例では、実行ファイルとして“C:¥tmp¥startup.bat”が登録されている。

3.1.4. タスクスケジューラ

このタスクスケジューラも永続化させる手段としてよく使われる。下記コマンドで、設定内容を出力できる。

コマンド例
schtasks /fo CSV /query /v

タスクスケジューラもデフォルトで多くのジョブが登録されている。出力する際には、「/fo CSV オプション」をつけて行う。このタスクスケジューラでも、PowerShell を利用する例が散見されるが、PowerShell の構文は長くなる性質があるため、このオプションをつけることで、文字切れを回避でき、設定内容を正しく出力することができる。

実行例
"ホスト名","タスク名","次回の実行時刻","状態","ログオン モード","前回の実行時刻","前回の結果","作成者","実行するタスク","開始","コメント","スケジュールされたタスクの状態","アイドル時間","電源管理","ユーザーとして実行","再度スケジュールされない場合はタスクを削除する","タスクを停止するまでの時間","スケジュール","スケジュールの種類","開始時刻","開始日","終了日","日","月","繰り返し: 間隔","繰り返し: 終了時刻","繰り返し: 期間","繰り返し: 実行中の場合は停止" "TESTPC","¥Adobe Acrobat Update Task","2018/01/28 12:00:00","不明","対話型/バックグラウンド","2018/01/27 17:11:44","0","Adobe Systems Incorporated","C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥AdobeARM.exe ","N/A","This task keeps your Adobe Reader and Acrobat applications up to date with the latest enhancements and security fixes","有効","無効","バッテリ モードで停止, バッテリで開始しない","INTERACTIVE","有効","72:00:00","スケジュール データをこの形式で使用することはできません。","ログオン時","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A"

スタートアップレジストリ同様に、実行するアプリケーションが指定されている。”次回の実行時間”、”前回の実行時刻”、そして、”実行するタスク”が、特に重要な項目だ。タスクスケジューラは、永続化設定以外にも、プログラムのリモート実行を行うことができるため、攻撃痕跡としても重要な調査対象となる。

なお、これらのタスクは、下記フォルダにジョブファイルとして配置されている。

フォルダパス
C:¥Windows¥System32¥Tasks

ファイル例
c:¥Windows¥System32¥Tasks のディレクトリ 2018/01/11 09:47   <DIR>   . 2018/01/11 09:47   <DIR>   .. 2017/11/16 12:13   4,464 Adobe Acrobat Update Task

```
2018/01/11  09:47             4,120 Adobe Flash Player Updater
```

ジョブファイルのタイムスタンプは、ジョブが生成、更新されたタイミングと推測できるため、情報収集時にはこのフォルダのファイル一覧も取得する。

3.2. 外部通信情報の取得

外部通信情報は以下の２種類から取得する。

- DNS キャッシュ
- ネットワーク接続情報

3.2.1. DNS キャッシュ

DNS キャッシュは、下記コマンドで出力する。

コマンド例
ipconfig /displaydns

ローカルドメインの名前解決も含めて出力される。このキャッシュは、OS 起動時にはクリアされてしまうので、取得タイミングには注意する。

実行例（抜粋）

```
ipconfig /displaydns
Windows IP 構成


    time.windows.com
    ----------------------------------------
    レコード名 . . . . . : time.windows.com
    レコードの種類 . . . : 1
    Time To Live  . . . .: 3462
    データの長さ . . . . : 4
    セクション . . . . . . . : 回答
    A (ホスト) レコード. . . : 52.163.118.68
```

「レコード名」が FQDN で、「 A （ホスト） レコード」が対応した IP アドレスである。

3.2.2. ネットワーク接続情報

ネットワーク接続情報は、下記のコマンドで出力する。

コマンド構文
netstat　オプション

netstat はネットワーク接続情報を出力する。取得時には、“-naob”、“-fao”のオプションを切り替えてそれぞれ取得する。“-naob”には管理者権限が必要なので、一般利用者権限の場合は、“-nao”で取得する。

コマンド構文（抜粋）
netstat -h プロトコルの統計と現在の TCP/IP ネットワーク接続を表示します。 NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [interval] -a　すべての接続とリッスン ポートを表示します。 -b　それぞれの接続またはリッスン ポートの作成に使われた実行可能ファイルを表示します。場合により、よく知られた実行可能ファイルが複数の独立したコンポーネントをホストすることもあり、この場合、接続またはリッスン ポートの作成に使われたコンポーネント群が表示されます。 この場合、実行可能ファイル名は下に [] で表示され、上には TCP/IP に到達するまで順に呼び出したコンポーネントが表示されます。 このオプションには時間がかかり、十分なアクセス許可がないとエラーが発生することに注意してください。 -f　外部アドレスの完全修飾ドメイン名 (FQDN) を表示します。 -n　アドレスとポート番号を数値形式で表示します。 -o　各接続に関連付けられたそれらを所有するプロセス ID を表示します。

実行例（抜粋）
C:¥Windows¥system32>netstat -naob アクティブな接続

```
  プロトコル  ローカル アドレス       外部アドレス           状態            PID
  TCP         0.0.0.0:135           0.0.0.0:0              LISTENING       864
  RpcSs
 [svchost.exe]
  TCP         0.0.0.0:445           0.0.0.0:0              LISTENING       4
 所有者情報を取得できません
  TCP         0.0.0.0:49408         0.0.0.0:0              LISTENING       644
 所有者情報を取得できません
  TCP         0.0.0.0:49409         0.0.0.0:0              LISTENING       1096
  EventLog
 [svchost.exe]
  TCP         0.0.0.0:49410         0.0.0.0:0              LISTENING       1492
 [spoolsv.exe]
```

“naob”は、プロセス ID を出力しているため、通信を行っているプロセスを特定することに役立つ。

実行例（抜粋）

```
C:¥Windows¥system32>netstat -fao
アクティブな接続

  プロトコル  ローカル アドレス       外部アドレス           状態            PID
  TCP         0.0.0.0:135           PC01:0         LISTENING       864
  TCP         0.0.0.0:445           PC01:0         LISTENING       4
  TCP         0.0.0.0:49408         PC01:0         LISTENING       644
  TCP         0.0.0.0:49409         PC01:0         LISTENING       1096
  TCP         0.0.0.0:49410         PC01:0         LISTENING       1492
```

“fao”は、FQDN の出力であるため、後の評価時において FQDN の調査時に役立つ。

ただし、C2 サーバとの通信を出力するという意味では、プロキシサーバを利用している環境の場合、PC から出力できるネットワーク接続情報には、プロキシサーバとの通信しか記録されない。調査対象の PC の外部通信環境によって、取得要否の判断をしてほしい。

3.3. 実行痕跡の取得

3.3.1. アプリケーションの実行痕跡

実行痕跡の情報収集は、下記の分類毎に取得する。

表 3.3-1 実行痕跡の情報収集方法

実行痕跡	内容と収集方法
Prefetch Files	C:¥Windows¥Prefetch フォルダに存在するファイルの一覧を取得する。要管理者権限。
最近使ったファイル	C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Windows¥Recent フォルダに存在するファイルの一覧を取得する。
最近利用した Office ドキュメント	C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Windows¥Office¥Recent フォルダに存在するファイルの一覧を取得する。
AppCompatCache	下記レジストリを取得する。 HKLM¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥AppCompatCache
UserAssist	下記レジストリを取得する。 HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥UserAssist
RunMRU	下記レジストリを取得する。 HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥RunMRU
TypedURLs	下記レジストリを取得する。 HKCU¥SOFTWARE¥Microsoft¥Internet Explorer¥TypedURLs

%USERNAME%にあたる箇所は、利用しているユーザー名が入る。インシデント検知時に行う調査という意味合いでは、現在利用しているユーザーでの情報収集が望ましい。しかし、複数ユーザーで利用している場合や、管理者ユーザーでのログインである場合は、調査対象 PC に保存されている「ユーザープロファイル」を確認し、情報収集の対象を決定する。HKCU はログイン中のユーザー分しか取得できないが、ファイル一覧であれば、ログインしていないユーザー分も取得できる。

3.3.2. 実行痕跡の収集

(1) Prefetch

Prefetch フォルダに生成されたファイルのファイル名とタイムスタンプから、実行しようとしたファイル名と実行時期に関する情報が把握できる。

フォルダパス
C:¥Windows¥Prefetch

このフォルダに生成される Prefetch ファイルは、繰り返し記録される性質のファイルであるため、下記のように取得することで、初めて実行した時間（ファイル作成日）と一番最近実行した時間（最終更新日）を推測することができる。

コマンド例（ファイル作成日でソート）
作成日でソートした一覧表示 dir /od /tc C:¥Windows¥PreFetch¥

コマンド例（最終更新日でソート）
最終更新日でソートした一覧表示 dir /od /tw C:¥Windows¥PreFetch¥

DIR コマンドオプション（抜粋）
DIR [ドライブ:][パス][ファイル名] [/A[[:]属性]] [/B] [/C] [/D] [/L] [/N] [/O[[:]ソート順]] [/P] [/Q] [/R] [/S] [/T[[:]タイムフィールド]] [/W] [/X] [/4] /O　　ファイルを並べ替えて表示します。 ソート順　N　名前順（アルファベット）　S　サイズ順（小さい方から） E　拡張子順（アルファベット）　D　日時順（古い方から） G　グループ（ディレクトリから）-　降順 /T　　どのタイム フィールドを表示するか、または並べ替えに使用するかを指定します。 タイムフィールド C　作成 A　最終アクセス W　最終更新

作成日と最終更新日でソートしたファイル一覧表示を例にあげる。

実行例（ファイル作成日でソート）
dir /od /tc C:¥Windows¥PreFetch¥

```
2015/11/10  17:24             9,778 CMD.EXE-4A81B364.pf
2015/11/30  13:14            15,424 DEFRAG.EXE-588F90AD.pf
```

実行例（最終更新日でソート）

```
dir /od /tw C:¥Windows¥PreFetch¥

2018/03/19  20:09            15,424 DEFRAG.EXE-588F90AD.pf
2018/03/22  19:13             9,778 CMD.EXE-4A81B364.pf
```

この実行例では、“CMD.EXE”を初回実行した時間は「2015/11/10 17:24」、直近で実行した時間は「2018/3/22 19:13」と推測される。

参考情報）pf ファイル

PreFetch フォルダに格納されている pf ファイルには下記の情報が記録されている。

- アプリケーションの名前
- 前回起動した時刻
- 今まで何回起動されたか
- 起動時に読み込むファイル名

pf ファイルを解析すると、より多くの実行に関する情報が入手できる。なお、この pf ファイルの作成は、サーバ OS ではデフォルトで無効となっている。

(2) 最近使ったファイル

ユーザープロファイル配下の¥AppData¥Roaming¥Microsoft¥Windows¥Recent フォルダに生成されたリンクファイルのファイル名から、開いたファイルを判断できる。エクスプローラー経由で開いたファイルは、このフォルダにリンクファイルが生成される。

フォルダパス

```
C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Windows¥Recent
```

ファイル例

```
C:¥Users¥user01¥AppData¥Roaming¥Microsoft¥Windows¥Recent のディレクトリ

2018/03/22  19:10    <DIR>          .
2018/03/22  19:10    <DIR>          ..
2018/03/22  19:10               674 Powershell_error.lnk
```

2018/03/22　18:24　　934 TEST-PPT.lnk 2018/03/22　18:25　　776 TEST-WORD.lnk 2018/03/22　19:10　　471 tmp.lnk

出力例から、ユーザー“user01”がこれらのファイルを開いたと推測できる。

(3)　最近使用した Office ドキュメント

ユーザープロファイル配下の¥AppData¥Roaming¥Microsoft¥Office¥Recent フォルダに生成されたリンクファイルのファイル名から Office で開いたドキュメントが判断できる。Word や Excel 等の Office ドキュメントによる不審ファイルが特定されていた場合、このフォルダを調べると、開いたかどうかを確認することができる。

フォルダパス
C:¥Users¥%USERNAME%¥AppData¥Roaming¥Microsoft¥Office¥Recent

ファイル例
C:¥Users¥user01¥AppData¥Roaming¥Microsoft¥Office¥Recent のディレクトリ 2018/03/22　18:25　<DIR>　　. 2018/03/22　18:25　<DIR>　　.. 2018/03/22　18:25　　1,165 Templates.LNK 2018/03/22　18:24　　1,051 TEST-PPT.LNK 2018/03/22　18:25　　1,056 TEST-WORD.LNK 2018/03/22　18:25　　905 デスクトップ.LNK

出力例では、ユーザー“user01”が、“TEST-WORD”というファイルを開いたことが記録されていることがわかる。

(4)　AppCompatCache

AppCompatCache は Application Campatibility Cache の略称で、次回のアプリケーション起動のためのキャッシュとして保持される値である。更新タイミングが OS シャットダウン時であること、記録されている時刻情報は実行時間ではなく、実行されたファイルのタイムスタンプであることなど、配慮する点はあるが、プリフェッチよりも多くの数を保持できるため、実行痕跡として非常に有用な記録である。

AppCompatCashe レジストリ
HKLM¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥AppCompatCache

AppCompatCache はクエリではなくエクスポートで、任意のファイル名を付与して取得する。

コマンド例
reg export "HKLM¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥AppCompatCache" 任意のファイル名.reg

このレジストリは、バイナリデータであるため、可読化が必要になる。次章で、解析するツールを紹介する。

(5) UserAssist

UserAssist には「エクスプローラー経由で実行されたファイル」が記録されている。このキーは、“HKEY_CURRENT_USER”、ユーザーに紐付くレジストリであることから、どのユーザーアカウントが実行したか、という判断に使うことができる。

UserAssist レジストリ
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥UserAssist

AppCompatCache と同様に、エクスポートで任意のファイル名で出力する。

コマンド例
reg export "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥UserAssist" 任意のファイル名.reg

この出力結果についても、評価するためには解析する必要がある。なお、本レポートではこの解析ツールについては、適切なものが発見できなかったため、取得までの記載としている。

(6) RunMRU

RunMRU もユーザーに紐付くレジストリーキーで、「ファイルを指定して実行」したファイル名が記録されている。

RunMRU レジストリ
HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥RunMRU

この値はクエリで取得する。

コマンド例
reg query ″ HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥RunMRU ″ /s

実行例
HKEY_CURRENT_USER¥Software¥Microsoft¥WIndows¥CUrrentVersion¥Explorer¥RunMRU a REG_SZ cmd¥1 MRUList REG_SZ cba b REG_SZ powershell¥1 c REG_SZ c:¥tmp¥startup.bat¥1

この出力例では、直近で「ファイルを指定して実行」したのは、“cmd”、つまりコマンドプロンプトであることがわかる。

(7) TypedURLs

TypedURLs はユーザーに紐付くレジストリーキーで、「InternetExplorer で実行されたURL」が記録されている。

TypedURLs レジストリ
HKCU¥SOFTWARE¥Microsoft¥Internet Explorer¥TypedURLs

この値はクエリで取得する。

コマンド例
reg query ″HKCU¥SOFTWARE¥Microsoft¥Internet Explorer¥TypedURLs″ /s

実行例
HKEY_CURRENT_USER¥SOFTWARE¥Microsoft¥Internet Explorer¥TypedURLs url1 REG_SZ http://www.microsoft.com/ url2 REG_SZ http://www.google.com/ url3 REG_SZ http://www.yahoo.co.jp/

この出力例では、最新の実行 URL は“http://www.microsoft.com”となる。

参考情報）PsExec の実行痕跡

PsExec はリモート実行ツールで、感染拡大によく利用される。このレジストリを確認することで、過去に実行したことがあるかを判断できる。

レジストリ
HKEY_USERS¥SID¥Software¥Sysinternals¥PsExec

クエリで確認することができる。

コマンド例
reg query "HKEY_USERS" /s /k /f PsExec reg query "HKCU¥Software¥Sysinternals¥PsExec" /s

実行例
C:¥Windows¥system32>reg query "HKEY_USERS" /s /k /f PsExec 検索の完了: 該当 0 件 C:¥Windows¥system32>reg query "HKCU¥Software¥Sysinternals¥PsExec" /s エラー: 指定されたレジストリ キーまたは値が見つかりませんでした

実行痕跡がない場合は、このような出力結果となる。

アプリケーションの実行痕跡として代表的なものをあげた。マルウェアの感染、ツールの実行という観点で、比較的痕跡が残りやすい箇所を取上げている。これらの痕跡はもちろん、マルウェアの実行痕跡だけが残るものではないので、マルウェア感染以外の確認でも利用できる。

3.4. 感染頻出箇所の取得

感染頻出箇所の特定フォルダを、サブフォルダとファイルの一覧を再帰的に、隠しファルを含めて表示させて、その結果をリダイレクトして保存する。再掲するが、下記のフォルダが感染頻出箇所だ。

表 3.4-1 感染頻出箇所

環境変数等	実フォルダ例
%TEMP%	C:¥Users¥%USERNAME%¥AppData¥Local¥Temp
%PROGRAMDATA% %ALLUSERSPROFILE%	C:¥ProgramData
%APPDATA%	C:¥Users¥%USERNAME%¥AppData¥Roaming
%LOCALAPPDATA%	C:¥Users¥%USERNAME%¥AppData¥Local
%PUBLIC%	C:¥Users¥Public

%USERNAME%にあたる箇所は、利用しているユーザー名が入る。インシデント検知時に行う調査という意味合いでは、現在利用しているユーザーでの情報収集が望ましい。しかし、複数ユーザーで利用している場合や、管理者ユーザーでのログインである場合は、調査対象 PC に保存されている「ユーザープロファイル」を確認し、情報収集の対象を決定する必要がある。(2.3(2)参照)

(1) ファイル名一覧の取得

ユーザープロファイルが保存されたフォルダと、それ以外の感染頻出場所のフォルダを隠しファイルを含めて一覧を収集する。ユーザープロファイル以下のフォルダはまとめて取得し、あわせてタスクスケジューラのジョブファイルやアプリケーションクラッシュ等 OS 上の主要なファイルが集められている C:¥Windows¥System32 も取得しておく。

表 3.4-2 取得すべきフォルダと含まれる環境変数

実フォルダ	含まれる環境変数
C:¥ProgramData	%PROGRAMDATA%
C:¥Users¥%USERNAME%¥Appdata	%APPDATA% %LOCALAPPDATA% %TEMP%
C:¥Users¥Public	%PUBLIC%
C:¥Windows¥System32	-

コマンド構文（抜粋）		
dir /a /r /s		
/A	指定された属性のファイルを表示します。	
属性	D ディレクトリ	R 読み取り専用
	H 隠しファイル	A アーカイブ

	S　システム ファイル	I　非インデックス対象ファイル
	L　再解析ポイント	-　その属性以外
/R	ファイルの代替データ ストリームを表示します。	
/S	指定されたディレクトリおよびそのサブディレクトリのすべての ファイルを表示します。	

マルウェアや不審ファイルは、隠し属性が与えられていたり、補助的なデータ保存場所である代替データストリームといった場所に隠されている場合がある。よって、ファイル名一覧を取得する際には、これらも表示させるオプションが必要である。

コマンド例
dir /a /r /s C:¥ProgramData
dir /a /r /s C:¥Users¥%USERNAME%¥AppData
dir /a /r /s C:¥Users¥Public
dir /a /r /s C:¥Windows¥System32

“/a”は、隠しファイルやシステムファイルも含めた全ての属性を表示し、“/r”は代替データストリームを表示させる。あわせて、“/s”でサブディレクトリ以下を表示させる。

代替データストリーム例
2017/12/06　19:46　　598,309 000062239.pdf
26 000062239.pdf:Zone.Identifier:$DATA

この例では、インターネットからダウンロードしたファイルに付加される“Zone.Indentifier”という代替データストリームが表示されている。

3.5. 情報収集コマンドの整理

本章で扱った情報収集コマンドを列挙する。収集対象の分類毎に解説したが、ここでは収集方法毎にまとめておく。

- レジストリ収集
- タスク収集
- ファイル収集
- ネットワーク情報収集
- ファイル一覧収集

(1)　レジストリ収集

レジストリ収集

```
# startup
reg query "HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce" /s
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run" /s
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Userinit"
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Shell"
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon¥Notify" /s
reg query "HKLM¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "AppInit_DLLs"
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Run" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnce" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunOnceEx" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServices" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥RunServicesOnce" /s
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥policies¥Explorer¥Run" /s
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "Load"
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Windows" /v "Run"
reg query "HKCU¥Software¥Microsoft¥Windows NT¥CurrentVersion¥Winlogon" /v "Shell"
reg query "HKU¥.DEFAULT¥Software¥Microsoft¥Windows¥CurrentVersion¥Run" /s
# services
reg query HKLM¥SYSTEM¥currentControlSet¥services /s
# startup folder
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders" /v "Startup"
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥Shell Folders" /v "Startup"
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders" /v "Startup"
reg query "HKLM¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥User Shell Folders" /v "Startup"
#RunMRU
reg query "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥RunMRU " /s
#TypedURL
reg query "HKCU¥SOFTWARE¥Microsoft¥Internet Explorer¥TypedURLs" /s
#PsExec
reg query "HKEY_USERS" /s /k /f PsExec
```

```
reg query "HKCU¥Software¥Sysinternals¥PsExec" /s
#AppCompatCache
reg export "HKLM¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥AppCompatCache" 任意のファイル名.reg
#UserAssist
reg export "HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥Explorer¥UserAssist" 任意のファイル名.reg
```

(2) タスク収集

タスク収集
schtasks /fo CSV /query /v

(3) リンクファイル確認

リンクファイル確認
PowerShell.exe -Command "& {filter Get-ShortCut{$shell=new-object -comobject△ WScript.Shell;return△ $shell.CreateShortcut($_);}$path=$env:APPDATA+¥"¥Microsoft¥Windows¥Start△ Menu¥Programs¥Startup¥";dir $path -r -include ¥"*.lnk¥" \|△ Get-ShortCut;$path=$env:PROGRAMDATA+¥"¥Microsoft¥Windows¥Start△ Menu¥Programs¥Startup¥";dir $path -r -include ¥"*.lnk¥" | Get-ShortCut;}"

本コマンド例は１行で実行する。各行末には半角空白の代替文字として△を記載した。

(4) ネットワーク情報収集

ネットワーク情報収集
ipconfig /displaydns netstat -nao netstat -naob netstat -fao

(5)　ファイル一覧収集

ファイル一覧収集
dir /a /r /s C:¥ProgramData
dir /a /r /s C:¥Users¥%USERNAME%¥AppData
dir /a /r /s C:¥Users¥Public
dir /a /r /s C:¥Windows¥System32
dir /od /tc C:¥Windows¥PreFetch
dir /od /tw C:¥Windows¥PreFetch

3.6. 活用例(バッチファイル)

これまで紹介した各コマンドが出力結果には、関連性を見ながら確認していく方が効率的なものがいくつかある。また、これらを一括に取得することで、「ある時点」の PC 状態を記録として出力することができる。各コマンドを一括取得するようバッチファイル化しておくことを推奨したい。その際、下記のような、PC の基本的な情報をあわせて取得しておくと、よりまとまった情報として保存することができる。例えば、PC をセットアップしたタイミングで取得しておくと、設定情報の保存という意味合いと次回取得時に差分を見つけるために活用できる。

表 3.6-1 同時に取得すべき項目

取得項目	コマンド例
システム情報	systeminfo
環境変数	set
現在ログオン中のユーザー情報詳細	whoami /all
WindowsUpdate 適用状態	wmic qfe list
日付と時間	wmic os get localdatetime

4. 評価

本章では、それらの収集した情報を評価する手順や着眼点を分類毎に解説する。後半は、ひとつの攻撃事例をもとに、それらの攻撃がどの場所にどのような痕跡を残したかを解説する。

4.1. 評価の基本的な手順

評価の基本的な手順を、図 4.1-1 に示す。

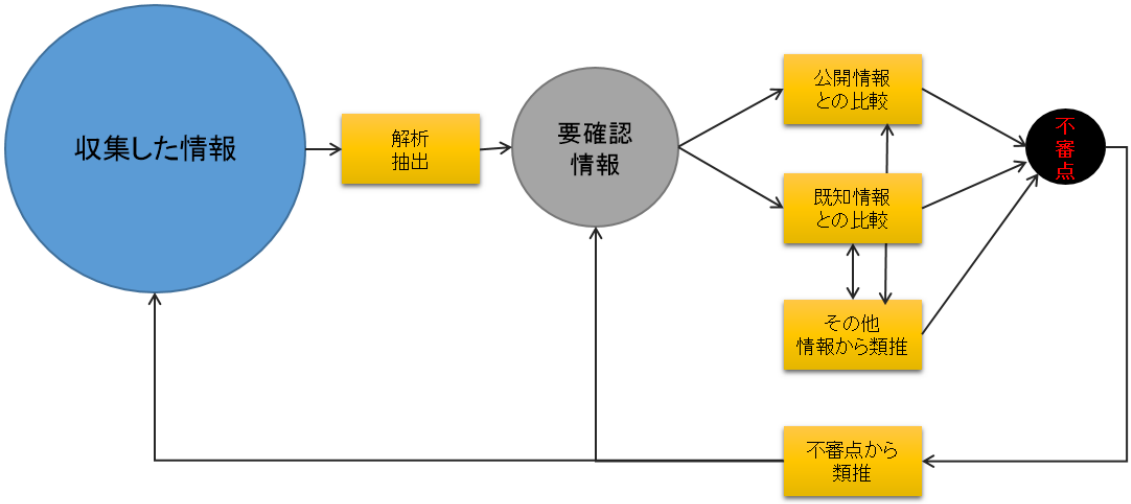


図 4.1-1 評価の基本的な手順

まず、収集した情報から、評価する対象にあわせて、解析、抽出し、要確認情報を取り出す。つぎに、その情報を３つの観点「公開情報との比較」「既知情報との比較」「その他の情報からの類推」で、比較・類推を行い、不審点を抽出する。そして、不審点の内容によっては、それらの不審点に関する痕跡等が、その他の収集情報や要確認情報に存在しないかを再度確認する。このような手順で、収集した情報から「不審点」を抽出していく。

図に示した各手順の概要と目的を**表 4.1-1** にまとめた。

表 4.1-1 評価手順の概要と目的

手順	概要	目的
解析/抽出	● 実行形式部分に注目し、感染頻出箇所を抽出する ● 解析し可読化する（必要な場合）	● 要確認情報を絞り込む
公開情報との比較	● 公開情報で正規情報を得る ● 公開情報で攻撃情報を得る	● 偽装を発見する ● インディケータを発見する
既知情報との比較	● 既知攻撃情報との類似点を抽出する ● 不審ファイル・攻撃ツールの有無を調べる	● インディケータを発見する
その他情報からの類推	● 実フォルダ確認して類推する ● 各種痕跡から類推する	● 関連する不審点を探す
不審点から類推	● 抽出できた不審点から類推する	● 関連する不審点を探す

これらの手順で補足すべき点をあげる。

(1) 類推から発見する不審点

評価手順に「類推」をあげているが、不審点が見つかった場合は、そこで得られた情報をもとに、他の収集結果を確認することで、あらたな不審点が見つかることが多々ある。マルウェアが感染するタイミングで複数の痕跡を残す場合があるからだ。具体的には、以下のような点に注目する。

- 永続化されている不審ファイルの実行痕跡が残っていないか
- 実行痕跡に見られた日付、時間とほぼ同時刻に更新されているフォルダやファイルはないか

これは、マルウェア感染後、攻撃者がなんらかの活動を行った場合でも同じことが言えるため、結果的に複数の不審点が見られることが多い。また、攻撃者はネットワークを経由して感染を拡大していくため、同一組織内で類似した不審点が見つかる傾向がある。組織内 PC を対象に調査をする場合は、これらの発見できた不審点を「同件」として調査することで、短期間に攻撃の範囲を把握するための目安にすることができる。

(2) 発見すべき不審点の例

評価において、発見すべき不審点の例を列挙しておく。

- 感染頻出箇所にある覚えのないファイルが永続化設定されており、正規ファイルではない可能性がある。
- 通信先の中に攻撃情報に紐付いた FQDN、IP アドレスがある。
- 不審なメールの添付ファイルを実行した痕跡が残っている。
- 不審なメールにあった URL を踏んだ痕跡が残っている。
- 攻撃事例で使われたファイルと同じ名前のファイルがあり、実行した痕跡がある。

このような評価が得られた場合は、マルウェア等に感染している可能性があると考えられる。より詳しい調査に進めることを検討すべきである。

4.1.1. 評価対象別の手順

評価対象別の手順を、**表 4.1-2** 評対象別実施内容に示す。基本的な実施手順を、評価対象の特性にあわせて実施する。

表 4.1-2 評対象別実施内容

評価対象	実施内容
永続化設定	● 感染頻出箇所の抽出 ● 既知攻撃類似点の抽出 ● 公開情報との比較 ● 実フォルダ確認
外部通信	● 通信先の抽出 ● 公開情報との比較
実行痕跡	● バイナリ解析（可読化） ● 感染頻出箇所の抽出 ● 実フォルダ確認 ● 不審ファイル・攻撃ツールの有無
感染頻出箇所	● 不審ファイル・攻撃ツールの有無

マルウェア感染の特性と収集する情報量を考慮し、永続化設定、実行痕跡をまず評価し、外部通信、感染頻出箇所と評価を進めることを推奨したい。

4.2. 永続化設定の評価

収集した永続化設定は、下記の４種類である。

- スタートアップレジストリ（スタートアップ起動プログラム）
- サービスレジストリ（サービス起動プログラム）
- スタートアップフォルダ
- タスクスケジューラ

収集結果からは、それぞれ何らかの「実行プログラム」が指定されていることが確認できる。これらの指定箇所に、「感染頻出箇所」や「偽装」、または「既知の攻撃事例との類似点」がないかを確認する。着眼点として主なものを**表 4.2-1** にあげる。

表 4.2-1 永続化設定の評価における着眼点

分類	着眼点	ポイント
感染	フォルダが感染頻出箇所である	実行ファイルが感染頻出箇所にある場合は、注意が必要だ。
偽装	サービス名、フォルダ名とファイル名に違和感がある	覚えのないソフトウェア名があり、指定されている実行ファイルのフォルダ名とファイル名、またはサービス名に違和感がある。例えば、ファイル名が Google 関連なのにフォルダ名が Google 関連ではない、サービス名が Adobe 関連なのに Google のようなファイル名がついているなど。
偽装	本来存在しないフォルダにファイルがある	WindowsOS 標準のプログラムを偽装したケースでよく使われる。本来は System32 にあるはずのプログラムが別の場所にある場合など。
偽装	ファイルに隠し属性がついている	隠し属性にすることで、デフォルトのエクスプローラー表示では見えなくする例があった。
既知	スクリプトが指定されている	スクリプトの種類は、javascript、vbscript、PowerShell がよく利用される。
既知	バッチファイルが指定されている	バッチファイルが指定されている場合も、内容確認は必要である。
既知	リンクファイルが指定されている	リンクファイルは、実行形式プログラムを隠蔽するのに適している。2017 年によく見られた PowerShell を使った攻撃でも、このリンクファイルで配置するものがあった。

既知	別プログラムを呼び出せるプログラムが指定されている。	直接プログラムを指定するのではなく、別のプログラムや DLL を呼び出せるプログラムが指定されている場合も注意する必要がある。 例）cmd.exe rundll32.exe regsvr32.exe mshta.exe PowerShell.exe これらの呼び出し先が、感染頻出場所のスクリプト類や、ネットワーク経由でダウンロードさせるような指定といった例があった。

以降は、実際の感染例をあげながら、どこが「不審な点」であるかを解説する。

4.2.1. スタートアップ起動プログラムの評価と感染例

(1) スタートアップ起動プログラム評価の手順

このレジストリはテキスト出力となり、量もそれほど多くは出力されない。しかし、非常によく使われる永続化の設定箇所なので、注意深く見る必要がある。永続化評価の着眼点にしたがって、情報収集結果から抽出、評価を行う。

(2) スタートアップ起動プログラム感染例

ここでは、感染例をもとに、どのような点が不審点かを解説する。

スタートアップ起動プログラムでの感染例		
HKEY_CURRENT_USER¥Software¥Microsoft¥Windows¥CurrentVersion¥Run		
taskeng	REG_SZ	″C:¥ProgramData¥taskeng.exe″
F3	REG_SZ	″C:¥ProgramData¥F3¥googleUpdate.exe″ 200 0
AYCais	REG_SZ	C:¥ProgramData¥AYCai¥AYCRunSC.exe

３つのプログラムが、HKEY_CURRENT_USER(HKCU)に指定されており、ユーザーがログインしたタイミングで、これらのプログラムが実行される。

さて、ここでの違和感をあげてみよう。ぱっと見た感じの違和感としては、この２つくらいだろうか？

- ３つとも感染頻出箇所の ProgramData フォルダ以下のサブフォルダにある
- ２つめの“googleUpdate.exe”は Google 関連を想像させるが、関係なさそうな名称である F3 フォルダにある

実行ファイル名を公開情報で検索してみると、おおよそ以下の情報が得られた。

- “taskeng.exe”は WindowsOS の正規プログラムで、タスクスケジューラの実行ファイル。C:¥Windows¥System32 フォルダに存在する。
- “googleUpdate.exe”は Chrome などをアップデートするための実行ファイルで、C:¥Program Files のサブフォルダに存在する。
- “AYCRunSC.exe”は、ESTsoft の ALYac というセキュリティソフトの実行ファイル。ESTSoft¥ALYac フォルダに存在する。

このような感染頻出箇所にあり、かつ覚えのないプログラム名であれば、フォルダ内容を確認する。以下にそれぞれのフォルダでのファイル一覧を示す。

ファイル例
C:¥ProgramData のディレクトリ 2014/04/29 17:47 62,464 taskeng.exe

ファイル例
C:¥ProgramData¥F3 のディレクトリ 2014/05/21 17:29 <DIR> . 2014/05/21 17:29 <DIR> .. 2013/08/29 10:50 116,648 googleUpdate.exe 2013/08/29 10:50 40,960 goopdate.dll 2013/08/29 10:50 108,628 goopdate.dll.map 2013/08/29 10:50 1,855 NvSmart.hlp 4 個のファイル 268,091 バイト

ファイル例
C:¥ProgramData¥AYCai のディレクトリ 2014/06/30 09:15 <DIR> . 2014/06/30 09:15 <DIR> ..

```
2013/08/29  10:50            47,628 itd
2013/08/29  10:50             3,072 ptl.aym
2013/08/29  10:50           118,087 ptl.ayx
2014/06/26  09:23                 8 rwncksspwsfxjhufje
               4 個のファイル             168,795 バイト
```

これらのファイル一覧からは、下記のようなことがわかる。

- “taskeng.exe”が置かれている C:¥ProgramData 直下にファイルがあるのは、違和感がある。
- “googleUpdate.exe”以外のファイルも Google 関連しそうではある。しかし、NvSmart.hlp というファイル名は、標的型攻撃で利用される PlugX の構成ファイルとしてあげられている例がある。
- “AYCRunSC.exe”が存在しない。また、用途の不明なファイルが生成されているのは違和感がある。

先の公開情報とあわせて整理すると**表 4.2-2** の評価になる。

表 4.2-2 不審ファイルとその理由

不審なファイル	不審な理由
taskeng.exe	● Windows 正規ファイルなのに ProgramData にあること ● ProgramData 直下にあること
googleUpdate.exe	● 本来の Program Files 以下にないこと ● F3 フォルダは Google 製品とは関連がないこと ● GoogleUpdate.exe の先頭の“G”が“g”になっていること ● NVSmart.hlp は既知の攻撃情報で PlugX の構成ファイルの可能性があること
AYCRunSC.exe	● 本来のフォルダパス ESTSoft¥ALYac にないこと ● 永続化設定しているのに、実行ファイルが存在しないこと

この例にあげた３つの永続化されたファイルは、標的型攻撃を行い感染させたマルウェアであると思われる。もちろん、ここまで記載した不審点という評価だけでなく、実際にファイルの調査など詳細の調査を行った結果から判断[6]した。

[6] 分析レポート 2016「長期感染の実態」参照のこと。

ポイント）偽装のテクニック例

１）細微な変更

Google などの著名なアプリケーションを模したものは、偽装のテクニックとしてよく使われるが、加えて、この例にあるような「大文字小文字」が一部だけ変更されていたり、“32”や“64”、“X”といった数字や文字を加えて、いかにもそれらしいファイル名にされていることがある。

２）元々あったフォルダを利用

マルウェアやツールの配置に、元々あったフォルダを利用する場合もある。例えば以下のようなフォルダだ。組織内で利用している共通のアプリケーションのフォルダが利用された例もある。

- 著名なアプリケーション
- ハードウェアのドライバー
- 開発系ソフトのフォルダ

参考情報）NSRL の利用

WindowsOS や主要 Office 製品等の構成ファイルのハッシュ値、ファイルサイズ等を収録した NIST[7]の“National Software Reference Library”がある。これらを利用するには、環境整備が必要であるが、実装すると非常に有用である。

名称	URL
National Software Reference Library(NSRL)	<https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl>

4.2.1. サービス起動プログラムの評価と感染例

(1) サービス起動プログラムの評価の手順

サービス起動プログラム設定レジストリは大量の収集結果となる。この結果に対しては、実行ファイルが指定されている箇所である、“ImagePath”の部分に着目し抽出していくことが効率的だ。よって、まずはこの ImagePath を抽出してから、永続化評価の着眼点にしたがって、情報収集結果から不審点がないかを評価していく。

[7] アメリカ国立標準技術研究所（National Institute of Standards and Technology）

(2)　サービスレジストリの感染例

サービスを使って永続化していた感染例を紹介し、不審点を解説する。

サービスレジストリ感染例
HKEY_LOCAL_MACHINE¥SYSTEM¥currentControlSet¥services¥hkcmd Module Service Type　REG_DWORD　0x110 Start　REG_DWORD　0x2 ErrorControl　REG_DWORD　0x0 ImagePath　REG_EXPAND_SZ　"C:¥ProgramData¥hkcmd¥hc.exe" 200 0 DisplayName　REG_SZ　hkcmd Module Service ObjectName　REG_SZ　LocalSystem Description　REG_SZ　hkcmd Module Service

これは“hmcmd Module Service”という名前でサービス登録されていた。ImagePath には、感染頻出箇所である ProgramData フォルダ配置のファイルが指定されている。

感染ファイル例
C:¥ProgramData¥hkcmd のディレクトリ 2013/07/08　18:52　<DIR>　. 2013/07/08　18:52　<DIR>　.. 2011/11/17　14:38　173,592 hc.exe 2011/11/17　14:38　2,560 hccutils.dll 2011/11/17　14:38　114,486 hccutils.dll.res 2016/04/09　15:57　9,467,820 NvSmart.hlp 4 個のファイル　9,758,458 バイト

これらの実行ファイル名、サービス名で公開情報を検索すると、下記のような情報が得られた。

- 実行ファイルの“hc.exe”はコンパックヘルプアプリケーション。
- “hkcmd Module”は Intel のグラフィックドライバに関するサービスの名称。正規パスは C:¥Windows¥System32 であり、hkcmd.exe が実行ファイルとされている。

この感染例でも、感染頻出箇所、正規ファイルとは異なるパス、偽装が見られる。そして、スタートアップレジストリの感染例と同じく既知の攻撃情報と合致するファイル名“NvSmart.hlp”が見られる。

ポイント）インディケータ

「公開情報での攻撃情報」という表現を使ったが、これらは、Indicator of Compromised（IOCs、インディケータ）と呼ばれる攻撃を示唆する情報のことである。セキュリティベンダーやリサーチャーが、標的型攻撃等のレポートを外部公開する中に、それらの攻撃で使われる要素を開示していることがある。インディケータには**表 4.2-3** のように様々な情報が扱われる。

表 4.2-3 インディケータの例

分類	内容
C2 サーバとの通信情報	● FQDN ● IP アドレス ● 通信ポート
攻撃メールの要素	● 件名 ● 送信者アドレス ● 送信ホスト ● 添付ファイル名 ● ハッシュ値
感染時に生成	● 永続化箇所 ● ファイル名 ● ハッシュ値

このような情報を収集しておくと、不審点の洗い出しや、感染をより早期に発見することにつながる場合がある。

参考情報）レジストリの最終書込み時刻

マルウェアの感染が発見されると、「いつ感染したか」は重要な調査項目になる。ここでは、その一例として、レジストリの最終書き込み時刻をあげる。レジストリには「最終更新日」が記録されており、下記手順で確認することが可能だ。

1）regedit で確認したいレジストリを探す
2）エクスポートでテキストファイルを指定する
3）エクスポートしたファイルには「最終書込み時刻」が記録されている

実行例
キー名: HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run クラス名: <クラスなし> 最終書き込み時刻: 2017/11/18 - 10:13 値 0 名前: Adobe ARM 種類: REG_SZ データ: "C:¥Program Files¥Common Files¥Adobe¥ARM¥1.0¥AdobeARM.exe"

この例では、AdobeARM.exe が自動起動設定になったのは、「2017/11/18 10:13」だと考えられる。これが、マルウェアの起動設定だったとしたら、この最終書込み時刻が、感染時間の有力な候補になる。

4.2.2. タスクスケジューラでの評価と感染例

(1) タスクスケジューラの評価の手順

タスクスケジューラの出力結果もかなり多い。実行形式ファイルが指定されているのは、“実行ファイル”のカラムであるが、その記載方法は様々であり、フルパスでない指定も見られる。紹介した情報収集の方法では、CSV 形式で出力しているため、抽出にあたっては、表計算ソフトでの処理も向いている。

(2) タスクスケジューラの感染例

タスクスケジューラを使った感染例では、以下の３つのパターンを紹介する。

- 永続化に利用
- マルウェア感染させるために１度だけ実行
- 一時的な作業のために１度だけ実行

永続化に利用された例をあげる。

タスクスケジューラ感染例①
"PC01","Updater","2017/08/27 9:00:00","準備完了","対話型のみ","2017/08/26 9:00:00","0","PC01¥","C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c "IEX

```
([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp
HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug)))
",",N/A",",N/A",",有効",",無効",",バッテリ",",モードで停止",",バッテリで開始しない
",",user01",",無効",",72:00:00",",スケジュール",",データをこの形式で使用することは
できません。",",毎日",",9:00:00",",2017/08/24",",N/A",",1",",日ごと",",",",N/A",",無効
",",無効",",無効",",無効"
```

この例では、毎日 9 時に PowerShell を使ったタスクが登録されている。HKCU¥Software¥Microsoft¥Windows¥CuurentVersion¥debug というレジストリに保存された値を、BASE64 でデコードした結果を PowerShell に引き渡す内容となっている。タスク内容を調べてみると、外部のサイトへ接続し、別のマルウェアをダウンロードするものとなっていた。

マルウェアの実行（マルウェア実行による感染）を行ったと思われる例をあげる。

タスクスケジューラ感染例②

```
",",PC02",",¥At1",",N/A",",準備完了",",対話型/バックグラウンド",",2017/11/23
12:30:00",",0",",N/A",",c:¥windows¥start.exe ",",N/A",",N/A",",有効",",無効",",バッテ
リ モードで停止, バッテリで開始しない",",AtServiceAccount",",有効",",72:00:00",",
スケジュール データをこの形式で使用することはできません。",",一度だけ
",",12:30:00",",2017/11/23",",N/A",",N/A",",N/A",",無効",",無効",",無効",",無効"
```

この例では、2017/11/23 の 12:30 に 1 度だけ、“c:¥windows¥start.exe”というファイルを実行する設定になっている。“At1”という名前は、タスク名を指定していない場合に自動的に付与されるジョブタイトルである。このタスクが実行されたことを契機に、この PC は感染したものと思われる。

ファイル例

```
C:¥Windows¥System32¥Tasks のディレクトリ

2017/11/17  04:53    <DIR>          .
2017/11/17  04:53    <DIR>          ..
2017/11/17  04:53             4,476 Adobe Acrobat Update Task
2017/11/23  12:28             1,246 At1
```

同タスクの生成されたジョブファイルであるが、ファイルの最終更新時間から、ジョブ登録した時間は「2017/11/23 12:28」であると推測される。

一時的な作業のためにタスク登録を行ったと思われる例をあげる。

タスクスケジューラ感染例③
"PC03","¥At1","N/A","準備完了","対話型/バックグラウンド","2017/12/22 16:03:00","0","N/A","taskkill /im powershell.exe /f","N/A","N/A","有効","無効","バッテリ モードで停止, バッテリで開始しない","AtServiceAccount","有効","72:00:00","スケジュール データをこの形式で使用することはできません。","一度だけ","16:03:00","2017/12/22","N/A","N/A","N/A","無効","無効","無効","無効"

このタスクは 2017/12/22 の 16:03 に“powershell.exe”のイメージ名を持ったタスクを停止するように指定されている。ジョブ登録の時間と思われる当該ファイルの最終更新日が 2017/12/22 の 16:02 であるところから、このタスク停止を、即実行するために登録したものと思われる。

ファイル例
2017/12/22　16:02　　1,326 At1

この組織では複数台の感染 PC が発見されたが、この例と同じようにタスクスケジューラによる「一時的な作業指示」の痕跡が多く残されていた。攻撃者の使う手法は被害組織内では、同じ傾向にあることを意味している。

タスクスケジューラはデフォルトの状態でもかなり多く登録されており、また、多くのソフトウェアでも自動的にアップデートを確認するようなタスクが作成もされている。そのため、初見で見分けるのは難しいが、実行するファイルのパスが感染頻出箇所であるもの、コメントがないものや、１度だけ実行となっているものは、いつ、どのような契機で登録されたタスクなのか確認することを推奨する。

また、タスクスケジューラのように大量の設定がありえるものは、セットアップや設定更新のタイミングで、設定を収集しファイルにしておくと、このような不審なタスクを発見したときの評価時に利用できる。

4.3. 通信先の評価

取得したDNSキャッシュとネットワーク接続情報からは、FQDNやIPアドレスという非常にわかりやすい形で出力される。しかし、マルウェアによる外部通信は正常通信にまぎれていることが殆どであるため、その評価は難しい。

4.3.1. DNSキャッシュの評価手順

このDNSキャッシュを使って、FQDNとその解決結果を抽出し、通信先の評価を行う。C2サーバの特性と考えられるのは、おおよそ以下の３つである。

- ローカルドメインには存在しない。
- 正規ドメインのサブドメインには存在しない。
- 過去に攻撃やマルウェア配布に利用された可能性がある。

これらの特性を利用すると、下記の手順での抽出となる。

１）ローカルドメインの名前解決とローカルネットワークアドレスを除く。
２）信頼できるドメイン名を除く。
３）抽出されたFQDNとIPアドレスを、攻撃情報を公開しているサイトで同件がないか確認する。

１）は自組織のネットワーク環境であり、比較的すぐ対応できるはずだ。２）については、準備作業が必要となる。下記のようなドメインは事前調査や事前検討を行い把握しておくとよい。

- 自組織で利用しているクラウドサービス等のドメイン名
- 自組織で利用してもよいと考えるドメイン名

前者は、業務に直結するような、契約、利用しているクラウドサービスやWebサービスなどを想定している。後者は一般的に利用してよいと思われるGoogle等の検索エンジン、採用しているPCやOS、アプリケーションのメーカーといったものや、各省庁や地方自治体、教育機関などを想定した。これらは、組織全体として把握しておくことが望まれる。

３）については、確認に利用する攻撃情報を公開しているサイトを、あらかじめリストアップしておき、１）、２）で取り除けなかったFQDNをこれらのサイトで同件がないかを確認する。そのような使い方が可能なサイトを**表 4.3-1**にあげる。

表 4.3-1 参考サイト

サイト名	概要
Threat Crowd	ドメイン名、IP アドレス、ファイルのハッシュ値等から、公開されている過去のマルウェア情報や関連情報が確認できる。 URL <https://www.threatcrowd.org/>
Passive Total	ドメイン名、IP アドレス、ファイルのハッシュ値等から、公開されている過去のマルウェア情報や関連情報が確認できる。WHOIS 履歴の参照も可能。 URL <https://community.riskiq.com/>
Virus Total	ファイルやウェブサイトの検査を行うウェブサイト。ドメイン名、IP アドレス、ファイルのハッシュ値等から検索することが可能。 URL <https://www.virustotal.com/>

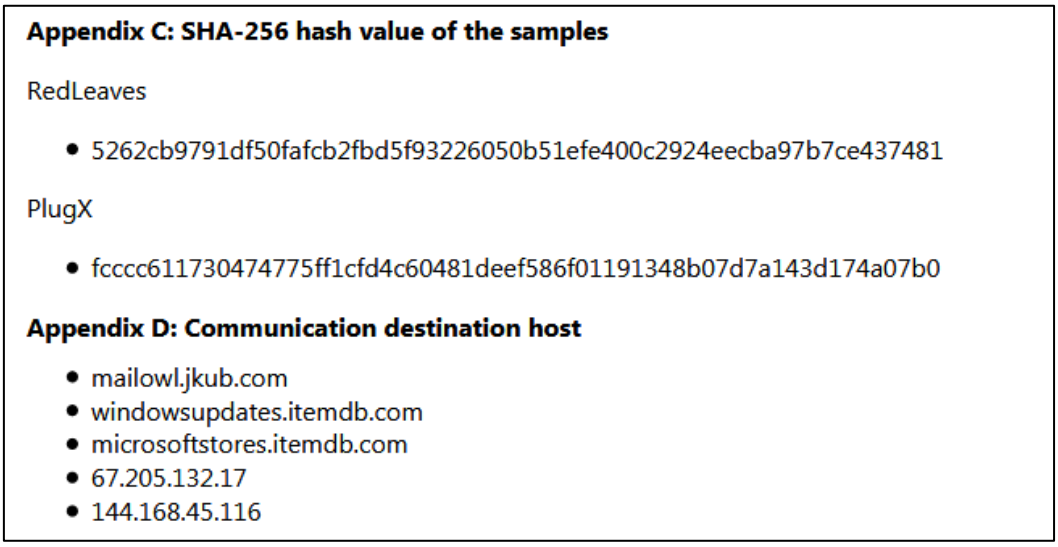
このような攻撃情報サイトでの検索結果に、攻撃情報が紐付いた場合でも、その情報には以下のような注意点がある。

- 攻撃情報は過去のものである。
- 攻撃期間が直近の場合は、まだ不審判定が出ていないことがある。
- VPS 等のサービスプロバイダーが利用しているドメインであると判断が難しい。

しかし、C2 サーバはある程度の期間、利用されていることが多いことから、過去の攻撃情報と紐付くという点で、不審な通信先であると考えていい。また、さらに Whois 情報の不審点や、名前解決された IP アドレスの過去の不審点をたどることで、さらなる不審情報が得られる場合がある。

ポイント）インディケータの活用

FQDN や IP アドレスでの判断に有効な情報源としては、Indicator of Compromise（インディケータ）がある。マルウェア感染の痕跡として、永続化箇所や生成されるファイルの情報もあるが、通信先に関する情報を得られることがある。このようなインディケータを収集しておくと、インシデント検知時やプロキシサーバや Firewall の調査時、または URL フィルターのブロック等に利用できる。



Appendix C: SHA-256 hash value of the samples

RedLeaves

- 5262cb9791df50fafcb2fbd5f93226050b51efe400c2924eecba97b7ce437481

PlugX

- fcccc611730474775ff1cfd4c60481deef586f01191348b07d7a143d174a07b0

Appendix D: Communication destination host

- mailowl.jkub.com
- windowsupdates.itemdb.com
- microsoftstores.itemdb.com
- 67.205.132.17
- 144.168.45.116

図 4.3-1　インディケータ例[8]

ポイント）ドメイン名の有効期間

ドメイン名や IP アドレスの利用は、有限性のものである。特にドメイン名については、登録、廃棄、更新、一時利用停止など、登録者側での操作で、状態が変化する。また、攻撃者が廃棄した後に、新たに取得した取得者が正規利用していることもある。よって、攻撃情報サイトに記載があるといっても、必ずしも「今現在、攻撃者が利用している」わけではない。

参考情報）ビーコンの停止

攻撃者がコントロールする RAT は、多くの場合、ビーコンと呼ばれる定期通信を C2 サーバと行う。しかし、この通信を発生させることが発見につながるケースがある。攻撃の過程において攻撃者は発見を回避するために、一時的に FQDN に対する A レコードを「0.0.0.0」や「127.0.0.1」とすることで、外部への通信を停止させることがある。明らかに外部のドメインであるのに、対する A レコードが「0.0.0.0」や「127.0.0.1」であった場合は、このような通信の抑制をしている可能性も考えられる。

[8]JPCERT/CC Official Blog Apr 03, 2017 “RedLeaves Malware Based on Open Source RAT” <http://blog.jpcert.or.jp/2017/04/redleaves---malware-based-on-open-source-rat.html>

4.3.2. ネットワーク接続情報の評価手順

収集したネットワーク情報からは、収集時の通信先が記録されている。よって、DNS キャッシュより、即時性の高い情報と言える。また、IP アドレスでリクエストしている通信が捕捉できる情報でもある。しかし、これも評価が難しい情報である。

抽出の仕方は、DNS クエリと同じ手順でよい。

1）ローカルドメインの名前解決とローカルネットワークアドレスを除く。
2）信頼できる IP アドレスを除く。
3）抽出された IP アドレスを、攻撃情報を公開しているサイトで同件がないか確認する。

しかし、FQDN に比べて視認性が低く、FQDN からの変換が必要であるため、2）は現実的ではないかもしれない。また、“naob”で取得していれば、取得した情報でプロセスを特定できるため、不審通信先が特定されている場合の評価に適している。

痕跡例
TCP 192.168.1.2:51183 [不審 IP アドレス]:1080 SYN_SENT 4740 [iexplore.exe]

例えば、この痕跡例では、不審通信先の IP アドレスから、この“PID 4740 iexplore.exe”のプロセスが関連していることが確認できた。

4.4. 実行痕跡の評価

4.4.1. 実行痕跡評価の手順

実行痕跡の情報収集結果は、ツールによる可読化が必要なものがある。評価の前には「ツールによる可読化作業」を行う。

表 4.4-1 可読化作業

実行痕跡	可読化の必要	必要な環境
Prefetch Files	×	-
AppCompatCache	○	Python
UserAssist	○	(参考のため記載)
RunMRU	×	-

TypedURLs	×	-

可読化作業に必要な解析ツールは、以降に紹介しているが、紹介したもの以外にも、数多く存在する。自組織の環境に適したツールを探して試行することも推奨する。

(1) AppCompatCache の可読化

AppCompatCache の解析には、下記ツールを利用する。

表 4.4-2AppCompatCache 解析ツールと入手先

ツール名	入手先
ShimCacheParser.py	URL<https://github.com/mandiant/ShimCacheParser>

前章で取得した AppCompatCache のレジストリを対象に解析する。解析ツールの構文は以下の通りだ。

解析ツール構文

```
ShimCacheParser.py -r 取得したレジストリ -o 出力結果のファイル名

usage: ShimCacheParser.py [-h] [-v] [-t] [-B] [-o FILE]
                          [-l | -b BIN | -m XML | -z ZIP | -i HIVE | -r REG]

Parses Application Compatibilty Shim Cache data

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose         Toggles verbose output
  -t, --isotime         Use YYYY-MM-DD ISO format instead of MM/DD/YY default
  -B, --bom             Write UTF8 BOM to CSV for easier Excel 2007+ import
  -l, --local           Reads data from local system
  -b BIN, --bin BIN     Reads data from a binary BIN file
  -m XML, --mir XML     Reads data from a MIR XML file
  -z ZIP, --zip ZIP     Reads ZIP file containing MIR registry acquisitions
  -i HIVE, --hive HIVE  Reads data from a registry reg HIVE
  -r REG, --reg REG     Reads data from a .reg registry export file
  -o FILE, --out FILE   Writes to CSV data to FILE (default is STDOUT)
```

解析ツール実行例
python ShimCacheParser.py -r AppCompatCache.txt -o ParsedAppCompatCache.csv [+] Reading .reg file: AppCompatCache.txt... [+] Found 32bit Windows 7/2k8-R2 Shim Cache data... [+] Writing output to ParsedAppCompatCache.csv...

この実行例では、前章で取得した AppCompatCache レジストリファイル（AppCompatCache.txt）を解析し、その結果を CSV ファイル（ParsedAppCompatCache.csv）に出力している。

実行結果例
Last Modified,Last Update,Path,File Size,Exec Flag 11/20/10 21:29:20,N/A,C:¥Windows¥system32¥wbem¥wmiprvse.exe,N/A,True 07/14/09 01:14:21,N/A,C:¥Windows¥system32¥ipconfig.exe,N/A,True 11/20/10 21:29:12,N/A,C:¥Windows¥system32¥cmd.exe,N/A,True 05/12/16 14:57:00,N/A,C:¥Windows¥system32¥gpscript.exe,N/A,True 11/20/10 21:29:13,N/A,C:¥Windows¥servicing¥TrustedInstaller.exe,N/A,True 11/20/10 21:29:20,N/A,C:¥Windows¥system32¥LogonUI.exe,N/A,True

4.4.2. 実行痕跡における着眼点

実行痕跡において、不審点を効率的に評価するためには、いくつかの評価すべき項目がある。例を**表 4.4-3** にあげる。

表 4.4-3 評価目的と検索対象キーワードの例

評価の目的	検索対象キーワード
不審ファイルを実行したかどうか	不審ファイル名、日付、時間
同組織で発見されたマルウェアに感染していないかどうか	マルウェアのファイル名、感染想定時期、感染拡大に利用されたツール名
マルウェアでよく使われる拡張子のファイルを実行したかどうか	偽装によく使われる拡張子
攻撃ツールが使われていないかどうか	攻撃ツール名のリスト

(1)　不審ファイル名、日付・時間での痕跡はないか

不審メールが特定されている場合は、添付ファイルを開いたかどうか、また不審ファイルが特定されている場合では、不審ファイルを実行してしまったかどうかを、実行痕跡を見ることで判断できることがある。

- 不審ファイル名
- 日付、時間

これらのキーワードで、実行痕跡の取得で出力したファイルを対象に、文字列検索を行う。出力結果のファイルの内容を考慮に入れながら、適時、検索の対象キーワードを変化させて、不審な実行痕跡がないか確認する。

ポイント）実行痕跡の時間情報

検索対象の「日付、時間」が「メールの受信日や時間」「不審ファイルを実行したと思われる日や時間」の場合は、実行痕跡毎に残っている情報が違うため、注意が必要だ。以下に各実行痕跡の時間情報に関する特性を記載する。

表 4.4-4 実行痕跡に残る時間情報

実行痕跡	時間情報の有無と留意点
Prefetch Files	最初に実行した時間と最新の実行時間
AppCompatCache	実行したファイルが持つ最終更新時間
UserAssist	最後に実行した時間
RunMRU	残らない
TypedURLs	残らない

実行痕跡が発見できた場合は、その実行時間が非常に重要なものだが、その実行痕跡に紐付く時間情報は、特定や推測が難しい。特に気をつけたいのが、多くの痕跡が残っていることが期待できる“AppCompatCache”だ。このレジストリが保持する時間情報は、実行されたファイルが持つ「最終更新時間」である。ファイルの種類によっては実行時間となるものがあるが、マルウェアの最終更新時間は意図的に変更されていることが多い。

ポイント）表記の違いを考慮

文字列検索での評価には、対象となる出力結果に現れる「表記の違い」を考慮する必要がある。代表的なものを**表 4.4-5** に３点あげる。日付の桁数、ファイル名、大文字小文字である。出力結果による違いも検索結果に反映できるよう考慮が必要だ。

表 4.4-5 文字列検索時の考慮事項

考慮事項	理由と対応
日付情報のゆらぎ	日付情報が 6 桁、8 桁のものや、出力の並びが、YYYY/MM/DD、MM/DD/YY と様々なので、出力内容を確認してから、それらにあわせてキーワードを指定する。
ファイル名の部分一致	Prefetch やアプリケーションクラッシュなど、ファイル名をもとに生成されるファイルを検索する場合は、ファイル名の部分一致となるよう指定する。
大文字小文字の判別	例えば、pf ファイルは、ファイル名を判別する箇所が大文字で出力されているため、実ファイル名とは変化する。大文字小文字を判別せずに検索する。

(2) 同組織で発見されたマルウェアに感染していないか

標的型攻撃の場合、被害組織で使われるマルウェアやツールは、共通している場合も多い。発見されたマルウェアやツールが、同一組織内の他の PC で、配置や実行がされていないかを見ることは大変重要である。

その一方で、攻撃者は活動時期によって、マルウェアやツールを変名や、ファイルの一部を変更して使う例もよく見られる。同じファイル名がなかったとしても、同時期、同じような痕跡箇所には注目して確認する必要がある。

参考情報）ハッシュ値

ファイルの同一性を評価するには、ハッシュ値の比較が適している。WindowsOS 標準コマンドでの取得方法と実行例を下記にあげる。もし、不審ファイルが見つかった場合、確実にセキュリティベンダー等へ伝えるには、ファイル名、ファイルサイズだけではなく、このハッシュ値をあわせて取得することが望ましい。

コマンド構文
certutil -hashfile ファイル名 引数

引数にアルゴリズムを指定する。デフォルトは SHA1 である。

コマンド実行例
C:¥Windows¥system32>certutil.exe -hashfile notepad.exe MD5 MD5 ハッシュ (ファイル notepad.exe): a4 f6 df 0e 33 e6 44 e8 02 c8 79 8e d9 4d 80 ea CertUtil: -hashfile コマンドは正常に完了しました。

(3)　偽装によく使われる拡張子の痕跡はないか

ここでは文字列検索だけなので、アイコン偽装の検出はできないが、拡張子であれば抽出可能だ。実行痕跡だけでなく、感染頻出場所フォルダでの不審ファイル抽出にも使える特性だ。ここでは、よく使われる Word 文書の“doc”ファイルでの例を**表 4.4-6** にあげる。

表 4.4-6 偽装によく使われる拡張子例（doc ファイルの例）

拡張子例	偽装
ファイル名.doc.exe	２重に拡張子を持つファイル。この例でいえば、拡張子表示していない場合でも、doc ファイルのように見える。マルウェアの場合、アイコンファイルも Word に見えるように偽装していることが多い。
ファイル名 cod.exe	RLO（Right-to-Left Override）による偽装のテクニック。この例でいえば、ファイル名は「ファイル名」exe.doc と表示される。もちろん、アイコンファイルもアプリケーションにあわせて偽装されていることが多い。
ファイル名.doc(長い空白).exe	ファイル名を長い空白を入れる偽装テクニック。長いファイル名は、後半が省略されて表示されるため、本当の拡張子である exe が隠されて、doc ファイルのように見える。

これらのファイル名が、実行痕跡や感染頻出箇所のフォルダなどに見つかった場合、標的型攻撃に遭遇し、かつ不審ファイルを実行した可能性が高いと思われる。

痕跡例
01/15/16 00:44:14 N/A C:¥Users¥User02¥Desktop¥[ファイル名]cod.exe N/A True

これは AppCompatCache の出力結果に残っていた痕跡例で、“[ファイル名]cod.exe”は、RLO を利用した[9]偽装 Word 文書である。実行結果が“True”であることから、実行に成功したものと考えられる。

[9] RLO のため、“[ファイル名]exe.doc”と表示するアプリケーションもある。

(4)　攻撃によく使われるツールの痕跡はないか

2.2.3「攻撃によく使われるツール」で記載した表を再掲する。これらのファイル名を使って、取得した結果から文字列検索を行う。実行痕跡の中には、実行時期が判別できないものもあるが、これらの痕跡が見つかれば、それが攻撃か管理者作業かを確認する必要がある。

表 4.4-7 攻撃によく使われるツール

ファイル名	機能など
Powershell.exe	WindowsOS 標準。高機能なスクリプト言語。近年特に攻撃での利用が多い。
wmic.exe	WindowsOS 標準。WindowsOS の管理基盤アーキテクチャ WMI（Windows Management Instrumentation）にアクセスするためのコマンドラインツール。
at.exe	WindowsOS 標準。スケジュール作成コマンド。 リモートコンピュータでのプログラム実行にも利用できる。
schtasks.exe	WindowsOS 標準。タスクスケジューラのタスクを操作するコマンド。リモートコンピュータでのプログラム実行にも利用できる。
Psexec.exe	Windows Sysinternals に含まれるリモートシステムでのプログラム実行ツール。組織内で感染を拡大する「横移動」によく使われる。
wce.exe	著名なハッキングツール。Windows Credential Editor の略で、ログオン中のパスワードを出力する。
Mimikatz.exe	著名なハッキングツール。メモリ上に保持されているアカウントの認証情報にアクセスし、各種クレデンシャル情報を奪取する。管理者権限の奪取によく使われる。様々な形態があり、PowerShell 版も存在する。
gsedump.exe	著名なハッキングツール。パスワードハッシュを出力する。

4.5. 感染頻出箇所の評価

ここまで解説してきた永続化設定や実行痕跡の評価において、感染頻出箇所が出てきた場合は、同フォルダの内容は確認をあわせて行ってほしい。しかし、永続化や実行痕跡で何もなかった場合も、念のため確認が必要だ。永続化設定がない例や、実行痕跡が残っていないがマルウェアや攻撃ツールが発見された例もあるからだ。

ここでは、そのような感染頻出箇所のみを評価するポイントについて解説する。

4.5.1. 感染頻出箇所における着眼点

感染頻出箇所においての評価すべき項目を**表 4.5-1** にあげる。

表 4.5-1 評価目的と検索対象キーワードの例

評価の目的	検索対象キーワード
攻撃ツールらしきものがないか	● 攻撃ツール名のリスト
情報窃取の痕跡らしきものがないか	● RAR や ZIP の覚えのない圧縮ファイルがないか ● コンピュータ名が使われたファイルがないか ● 組織名やドメイン名、ActiveDirectory のドメイン名が使われたファイルがないか

(1) 攻撃ツールらしきものがないか

実行痕跡の評価であげたよく使われる攻撃ツールのうち、WindowsOS 標準でないものが存在しないかどうかは確認することを推奨する。なお、これらのファイル名は一部を改変している例も散見されるため、それらを考慮して、攻撃ツールにファイル名が近いものを抽出することで不審なファイルが見つかる可能性もある。

(2) 情報窃取の痕跡らしきものがないか

不審ファイルの存在や攻撃ツールの実行痕跡が存在した場合、情報窃取の可能性が考えられる。攻撃者が外部へ情報を持ち出す場合、圧縮ファイルで行うことがあるとされている。このようなことから、RAR、ZIP といった圧縮ファイルの存在を確認することを推奨する。また、それらのファイル名に、コンピュータ名、被害組織の名称やドメイン名、ActiveDirectory のドメイン名をあらわす単語などが付与されている例も散見される。
このようなファイルには、パスワードがかかっていることもある。

参考情報）アプリケーションのクラッシュ情報

もし、実行痕跡の中でなんらかの「攻撃痕跡」があった場合、WindowsOS 標準機能で取得されるクラッシュ情報にも記録があるかもしれない。マルウェアやツールが実行環境と合わずにクラッシュしたと思われる痕跡が残っていることが時々あるからだ。

アプリケーションがクラッシュすると、下記フォルダにその実行ファイル名を含んだ名称のフォルダが生成され、各種情報が保存される。

フォルダパス
C:¥ProgramData¥Microsoft¥Windows¥WER¥ReportArchive

そして、これらのフォルダのタイムスタンプは、実行された日付である可能性が高いので、攻撃者の活動日付として重要な痕跡と言える。ただし、ファイル名は実行プログラム名が記録されるため、インジェクション等のテクニックを使われた場合は判別しにくいこともある。

痕跡例
2014/05/23　12:55　　<DIR>
AppCrash_msbridge.exe_6bb552452dab906c811ef559b45cb7fc369ef7a7_0123dbdd
2014/05/16　10:08　　<DIR>
AppCrash_msbridge.exe_6bb552452dab906c811ef559b45cb7fc369ef7a7_115fec41
2014/05/15　19:19　　<DIR>
AppCrash_msbridge.exe_6bb552452dab906c811ef559b45cb7fc369ef7a7_78d6c27e

この痕跡例から、2014/5/15、2014/5/16、2014/5/23 に実行したことが推測される。

4.6. 攻撃事例と痕跡

本章の最後は、ある攻撃事例にもとづき、攻撃遷移とそれらの痕跡がどこに残っていたかを解説する。PowerShell を使った攻撃であったため、不審ファイルは存在せず、動作の多くはイベントログにしか残っていなかった。しかし、感染の契機や永続化については、実行痕跡として残されていた。

4.6.1. 攻撃事例概要

この事例での攻撃遷移は以下の 6 段階があった。

1）攻撃メール受信
2）添付ファイル（マルウェア）実行
3）PowerShell コードを記載した hta ファイルをダウンロードし実行
4）PowerShell の RAT をリモートスクリプト実行
5）公開リポジトリから PowerShell のリモートスクリプト実行

6）Powershell の RAT を永続化

これらの攻撃遷移と痕跡箇所を図 4.6-1 に示す。

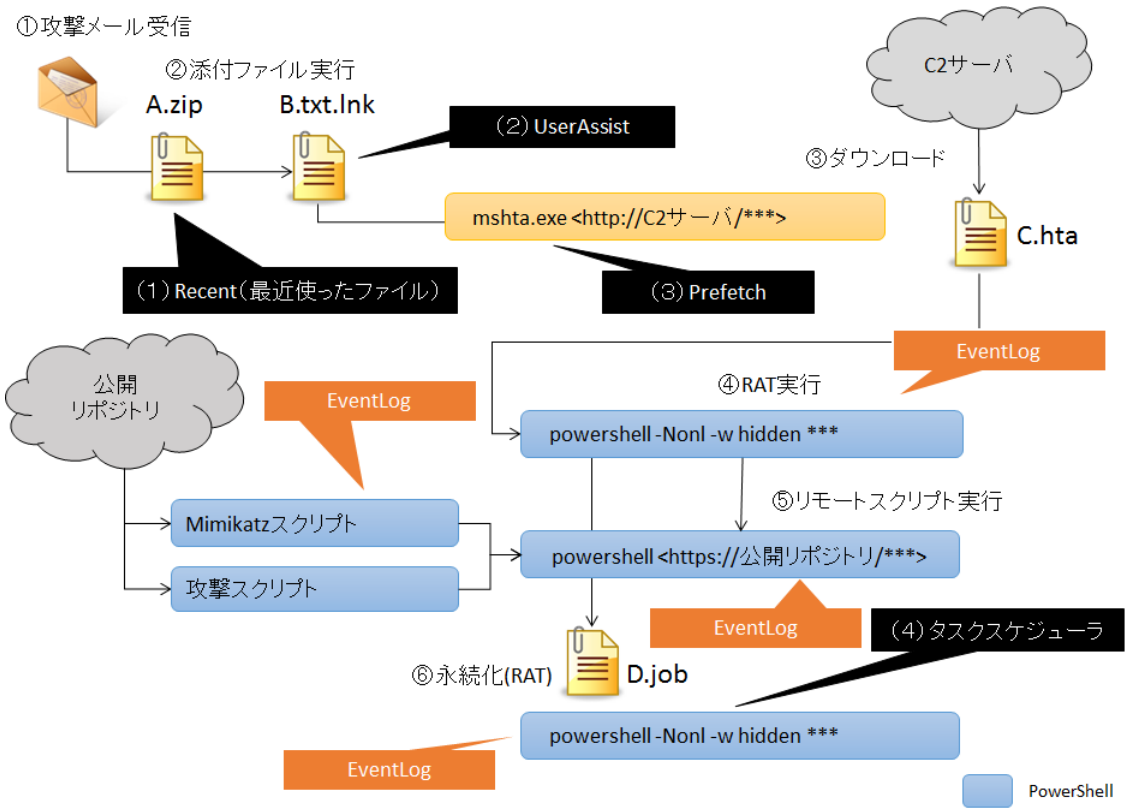


図 4.6-1　攻撃遷移と痕跡箇所

黒の箇所が、本レポートで扱っている「実行痕跡」と「永続化設定」にあたる。便宜上、図の表記と本文のファイル名は下記のように対応している。

- A.zip ：　[添付ファイル名].zip
- B.txt.lnk ：　[添付ファイル名].txt.lnk
- C.hta ：　-
- D.job ：　Updater

4.6.2.　痕跡

各箇所に残された痕跡を示す。読み取れる内容と、攻撃痕跡との関連を解説する。

(1) Recent(最近使ったファイル)

痕跡例
C:¥Users¥User01¥AppData¥Roaming¥Microsoft¥Windows¥Recent のディレクトリ 2017/08/24 16:44 647 [添付ファイル名].lnk

Recent フォルダに、マルウェアが仕込まれた ZIP ファイルの [添付ファイル名]がついた.lnk ファイルがあったため、当該 PC で ZIP ファイルを開いたと推測される。なお、lnk ファイルの内容を分析すると、当該 ZIP ファイルを 2017/8/24 の 16:39 に開いたことが記録されていた。(lnk ファイルの最終更新時間と異なる)

(2) UsersAssist(参考)

痕跡例
08/24/17 16:40:39,C:¥¥Users¥¥User01¥¥AppData¥¥Local¥¥Temp¥¥Temp1_[添付ファイル名].zip¥¥[添付ファイル名].txt.lnk

UserAssist に当該時間、user01 で当該ファイルが実行されたことが記録されていた。

(3) PreFetch

痕跡例
2017/08/24 16:40 29,938 MSHTA.EXE-A970B441.pf

最初のマルウェアが仕込まれた、“B.txtlnk”ファイルから通信が発生し、ダウンロードされた“C.hta (HTMLApplication)”ファイルを実行した“mshta.exe”が Prefetch に記録されていた。

(4)　タスクスケジューラ

感染例
"PC01","Updater","2017/08/27 9:00:00","準備完了","対話型のみ","2017/08/26 9:00:00","0","PC01¥","C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c "IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug))) ","N/A","N/A","有効","無効","バッテリ","モードで停止","バッテリで開始しない","user01","無効","72:00:00","スケジュール","データをこの形式で使用することはできません。","毎日","9:00:00","2017/08/24","N/A","1","日ごと","","N/A","無効","無効","無効","無効"

毎日 9 時に PowerShell を使ったタスクが登録されている。レジストリ”HKCU¥Software¥Microsoft¥Windows¥CurrentVersion¥debug”に保存された値を、Base64 で復号化した文字列を PowerShell で実行するタスクとなっている。そして、レジストリの値を復号化したところ、C2 サーバへ通信するものだった。

なお、その通信においてダウンロードされていたスクリプトは、外部公開されている PowerShell による RAT と同じ内容であった。

ファイル例
2017/08/24　23:14　　3,896 Updater

そして、このタスクスケジューラが登録または更新されたのは、2017/8/24 の 23:14 であると推測される。

このように、マルウェア感染後には複数の痕跡や攻撃者の活動痕跡を残している場合がある。

参考情報）Windows10 での PowerShell ログ

PowerShell の実行記録は、Windows7 のデフォルトである Ver2.0 では詳細は取得できないが、Ver5.0 の Windows10 では、デフォルトの設定でリモートコマンドの実行は「警告」としてイベントログが記録される。

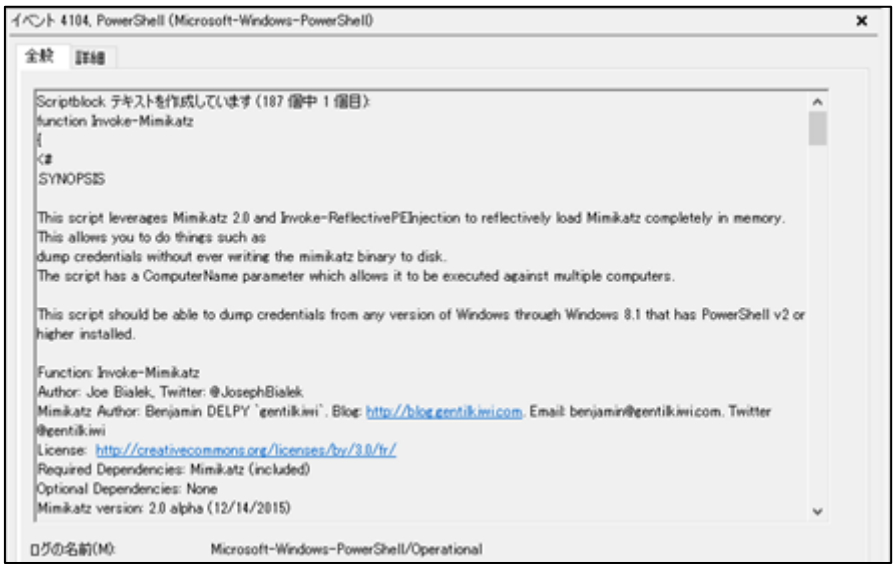


図 4.6-2 イベントログ出力例

この例では、187 個のスクリプトブロックに分けて実行されていた。

5. おわりに

J-CRAT チームが運営している「標的型サイバー攻撃特別相談窓口」を開設してから、数年経つが、その相談は変わらず多く寄せられている。不審ファイルの開封から、PC からの不審通信の発見、外部機関からの不審通信の警告、更にその中には、残念ながら深刻な状況となっているケースや、長期間に渡って攻撃にさらされていたケースもみられる。

この標的型攻撃への対策は、様々なツールを駆使した技術的な対策や、攻撃の初弾を受ける前線となるユーザーの教育や訓練等の人的対策や、運用管理面での対策等が行われている。しかし、いずれも完全ではなく、こうした中で重要なのは、懸念される操作や行為をした、不審な動きを検知した、といった操作ミスや攻撃の兆候を把握した際に、できるだけ早急な調査と対応の手を打つことが、長期感染やシステム内での蔓延や最終的な重大被害等を回避、低減する上で、非常に重要となる。

本レポートでは、インシデントの把握／発生時の対応にあたって、実施される調査、分析の全体像を提示している。調査、分析の目的は、攻撃の有無、感染の有無、被害範囲の把握、対策の有効性の検証などであり、システム（PC、サーバ、ネットワーク）や情報のどの箇所にどのような調査をするかを解説している。

その中でも、初動として、最初にとりかかるのが、感染の有無を検証する作業で、攻撃の入口となる PC に残された挙動や通信の記録の調査となる。それを、本レポートの読者自らが実施できるように、その手順と記録の解読方法を詳細に解説している。この調査は、J-CRAT でのレスキュー活動の初動で実践している調査の一部である。この結果から、次の行動（幹部への報告、調査の継続、外部ベンダーへの委託など）に移る判断が可能となり、被害の拡大抑止に向けた対応がとれるようになる。

例えば、システム管理者のあなたは相談者の PC から、不審ファイルの実行痕跡を見つけることができた。残念ながら実行してしまっていたが、永続化で不審点は見つからなかったので、次は DNS キャッシュとプロキシサーバの通信ログから通信先を洗い出して、不審通信先の調査をすることにした・・・

あるいは、感染の確認と内部や外部への通信ログの挙動が確認され、組織内への感染が広がっている懸念が高いと判断した。幹部への報告、予算稟議を通してセキュリティベンダーへの早急な調査依頼を出すこととした・・・

本レポートによって、各組織のインシデント対応能力、特に初動、攻撃の上流での調査能力の向上に繋がり、結果として、長期感染や重篤な被害が引き起こされるのを回避できるようになることを期待している。

また、本レポートで紹介した情報収集結果は、ある意味、PC の「現在の状態」を出力するものである。これは、インシデント検知時にのみ有効なものではなく、PC のセットアップ時に取得しておくことで、「変更前の状態」が保存できる。これは不審検知時には差分を確認する材料となる。さらに定期的に取得することで、健康診断の役割を果たすものにもなりえると考えている。

さて、最後は情報提供のお願いをしたい。

日々の「標的型サイバー攻撃特別相談窓口」「レスキュー活動」を通して、標的型攻撃への有効な対抗策のひとつは攻撃情報の共有であることを強く感じている。もう少し早く情報共有していれば、ひょっとしたら防げた事案ではなかったかと思う事も多々ある。

本当の攻撃は、攻撃者しか知らない。それらを知る手がかりは、標的型攻撃の標的となった組織に残された攻撃情報としてのみ存在する。それらの攻撃情報を確実に把握し、分析することは、我が国におけるサイバードメインアウェアネスの一助となると考えている。もちろん、それは標的型攻撃に対する我が国の強靭性を高めることにつながるものだ。各組織において、標的型攻撃を検知した場合には、「標的型サイバー攻撃特別相談窓口」への相談、または各種情報提供を是非お願いしたい。

サイバーレスキュー隊（J-CRAT）技術レポート 2017
インシデント発生時の初動調査の手引き
～WindowsOS 標準ツールで感染を見つける～

［発　　行］2018 年 3 月 29 日
［著作・制作］独立行政法人情報処理推進機構　技術本部　セキュリティセンター
［執　筆　者］釜谷誠　岩井拓　伊東宏明　青木眞夫　金野千里