アンラボ・セキュリティレター

Press Ahn

2015.6　Vol.18

身代金を支払わず復元できるランサムウェア

TeslaCrypt ランサムウェアの詳細分析

身代金を支払わず復元できるランサムウェア

PCに保存されたファイルを暗号化した後、復元することを条件に金銭的な対価を要求するランサムウェア（ransomware）が韓国でも相次いで発見され、ユーザーの不安が高まっている。最近ゲーム関連のファイルを狙うランサムウェア「TeslaCrypt」または「Tescrypt」も登場。幸い韓国ではまだ被害例が報告されていないが、実は身代金を払わなくともTeslaCryptランサムウェアに感染したファイルを復元できることが確認された。今回のコラムではTeslaCryptランサムウェアの攻撃方法と、感染したファイルの復元方法を分析する。今後のランサムウェア対策に活用していただきたい。

4月、ハングル版が発見されて大きく話題になったCryptoLockerランサムウェアは、explorer.exeやsvchost.exeなどの通常プロセスにコードをインジェクション（injection）する手法を使用した。TeslaCryptランサムウェアの場合は、%APPDATA%パスにマルウェアを作成・実行させることで感染する。TeslaCryptランサムウェアがファイルを暗号化するプロセスは次の通りだ。感染すると、まずユーザーPCには「.ecc」という名のファイルのみ存在するようになる。

normal.jpg→normal.jpg（暗号化）→normal.jpg.ecc（コピー）→normal.jpg（削除）

TeslaCryptランサムウェアの詳細分析

TeslaCryptランサムウェアの感染方法と動作フローを見てみよう。

1.ファイルとレジストリの作成

（1）ファイル作成

%APPDATA%パス（CSIDL_APPDATA）に右のファイルが作成される。

● パス: %APPDATA%: C:¥Documents and Settings¥<username>¥Application Data

- ランダム名.exe（例：asoddjv.exe）
- help.html
- log.html：感染ファイルのリスト情報
- key.dat：ファイル復号化に使用されるKEYファイル

PCのデスクトップパス（CSIDL_DESKTOPDIR）に以下のファイルが作成される。

- CryptoLocker.lnk : %APPDATA%asoddjv.exeファイルのショートカット
- HELP_TO_DECRYPT_YOUR_FILES.bmp : 警告ウィンドウ（「v4」）のイメージファイル
- HELP_TO_DECRYPT_YOUR_FILES.txt : 警告ウィンドウ（「v4」）のテキストファイル

(2) レジストリ登録

自動実行のためレジストリを登録する。

● レジストリ : HKCU¥ Software¥ Microsoft¥ Windows¥ CurrentVersion¥ Run
  - 値の名前 : crypto13 - 固定
  - 値のデータ : C : ¥ Documents and Settings¥<username>¥ ApplicationData¥ asoddjv.exe - 可変

2. Mutex作成

TeslaCryptランサムウェアは、次の固定Mutex名を使用する。

- System1230123

3. Volume Shadow削除

感染作業を実行する前に、次のコマンド（Volume Shadow copy Service（VSS）Admin）によって、すべてのウィンドウの復元イメージを使用不可にする。

- vssadmin delete shadows /all

4. 感染対象のドライブおよびファイル

TeslaCryptランサムウェアはドライブ名に関係なく、固定ドライブ（DRIVE_FIXED）のみ感染対象にするため、リムーバブルドライブやネットワークドライブについては感染対象から除外される。またドライブに保存されたファイルの中で、[表1]のような拡張子を持つファイルを感染対象とする。この中で、arch00、Day Profile、forge、mcgame、rgss3aなどがゲーム関連のファイルである。

.7z	.bkp	.sb	.vpk	.DayZProfile	.pak	.crt	.crw	.wpd	.odc
.rar	.qic	.fos	.tor	.rofl	.big	.cer	.bay	.dxg	.odm
.m4a	.bkf	.mcgame	.psk	.hkx	.unity3d	.der	.sr2	.xf	.odp
.wma	.sidn	.vdf	.rim	.bar	.wotreplay	.x3f	.srf	.dwg	.ods
.avi	.sidd	.ztmp	.w3x	.upk	.xxx	.srw	.arw	.pst	.odt
.wmv	.mddata	.sis	.fsh	.das	.desc	.pef	.3fr	.accdb	
.csv	.itl	.sid	.ntl	.iwi	.py	.ptx	.dng	.mdb	
.d3dbsp	.itdb	.ncf	.arch00	.litemod	.m3u	.r3d	.jpe	.pptm	
.sc2save	.icxs	.menu	.lvl	.asset	.flv	.rw2	.jpg	.pptx	
.sie	.hvpl	.layout	.snx	.forge	.js	.rwl	.cdr	.ppt	
.sum	.hplg	.dmp	.cfr	.ltx	.css	.raw	.indd	.xlk	
.ibank	.hkdb	.blob	.ff	.bsa	.rb	.raf	.ai	.xlsb	
.t13	.mdbackup	.esm	.vpp_pc	.apk	.png	.orf	.eps	.xlsm	
.t12	.syncdb	.001	.lrf	.re4	.jpeg	.nrw	.pdf	.xlsx	
.qdf	.gho	.vtf	.m2	.sav	.txt	.mrwref	.pdd	.xls	
.gdb	.cas	.dazip	.mcmeta	.lbf	.p7c	.mef	.psd	.wps	
.tax	.svg	.fpk	.vfs0	.slm	.p7b	.erf	.dbfv	.docm	
.pkpass	.map	.mlx	.mpqge	.bik	.p12	.kdc	.mdf	.docx	
.bc6	.wmo	.kf	.kdb	.epk	.pfx	.dcr	.wb2	.doc	
.bc7	.itm	.iwd	.db0	.rgss3a	.pem	.cr2	.rtf	.odb	

[表1] 感染対象ファイルの拡張子リスト

このとき、次の3つの条件に該当するファイルは感染対象から除外される。

-ファイル拡張子： 拡張子が「.ecc」ファイル形式を除く

-ファイル名： ファイル名に「Temporary」が含まれる場合を除く

-ファイルサイズ： 0x10000000バイト（約260MB以上）より大きいファイルを除く

感染されたファイルを復元する

TeslaCryptランサムウェア感染時に作成されるファイルの構造を分析し、ファイルを復元する方法を見てみよう。4月、海外のセキュリティベンダーもブログにてTeslaCryptランサムウェアに感染したファイルの復元方法を紹介している。

1.ファイル復元のキーファイル「key.dat」

%APPDATA%パスに作成されるデータファイルの中で、「key.dat」は暗号化されたファイルを復元するのに必要なAESキー（Key）情報を含んでいる。

[図1]は、TeslaCryptランサムウェアに感染すると表示される画面で、これによるとユーザーPCは「RSA-2048」方式で暗号化されており、復元に必要なPrivate Keyは、サーバーに保存されている。しかし実際にはRSA方式ではなくAES方式で暗号化されており、CBC（Cipher Block Chaining）モードが使用されていた。

[図1] TeslaCryptランサムウェア感染に関するお知らせウィンドウ

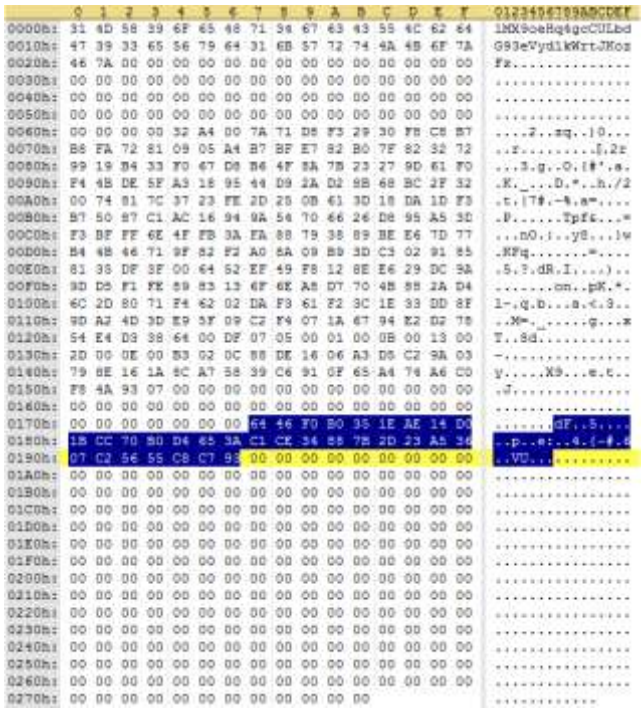
TeslaCryptランサムウェアで使用した暗号化方式は、[図2]のPythonサンプルコードのような形で構成される。

左記のサンプルコードで赤の「This is a key123」が最初のキー（AES Key）を表し、青の「This is an IV456」が第2キー（Initialization Vector）として使用されている。TeslaCryptランサムウェアが作成したkey.datファイルには、上記の赤で表示された「This is a key123」情報が存在する。

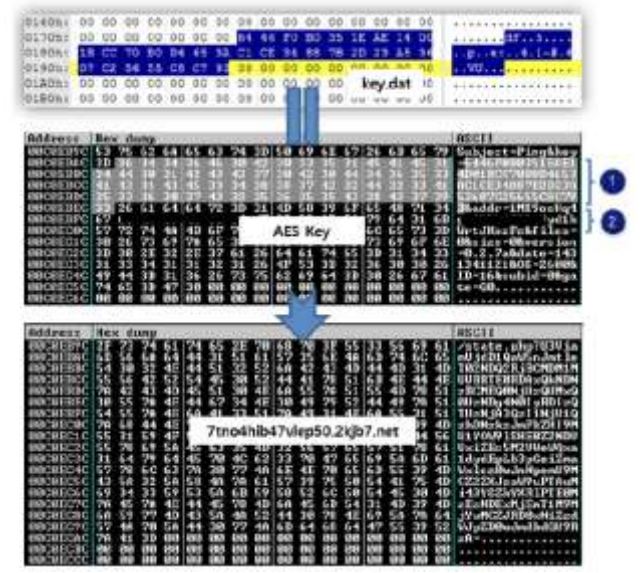
```
>>> from Crypto.Cipher import AES
>>> obj = AES.new('This is a key123', AES.MODE_CBC, 'This is an IV456')
>>> message = "The answer is no"
>>> ciphertext = obj.encrypt(message)
>>> ciphertext
'₩xd6₩x83₩x8dd!VT₩x92₩xaa`A₩x05₩xe0₩x9b₩x8b₩xf1'
>>> obj2 = AES.new('This is a key123', AES.MODE_CBC, 'This is an IV456')
>>> obj2.decrypt(ciphertext)
'The answer is no'
```

[図2] TeslaCryptランサムウェアの暗号化サンプル

また青で表示された「This is an IV456」情報に該当する初期化ベクトル（Initialization Vector、以下IV）値は、感染ファイルの最初の16バイトから得られる。正確には「key.dat」ファイルのオフセット（0x177~0x197）までの 0x20バイト値をSHA256に変換する過程を経て作成された値が、実際のAESキー（256-bits）として使用されている。



[図3] key.datファイルのAESキー（青い部分）



[図4] AESキー値の送信プロセス

（1）AESキー（Key）

TeslaCryptランサムウェアはkey.datファイルに存在するAESキー値（0x20バイト）と感染ファイルごとに異なるIV値（0x10）を通じて復元できる構造だ。[図3]は、key.datファイルの内容を示し、青く塗りつぶされた部分がAESキーに当たる0x20バイトを示す。

この値は、ファイルの開始から0x177オフセットだけ離れた位置に存在し、全体のファイルサイズは0x27Cバイトで固定される。

このように作成されたAESキーの値は、[図4]のように暗号化され攻撃者のサーバーに送信される。

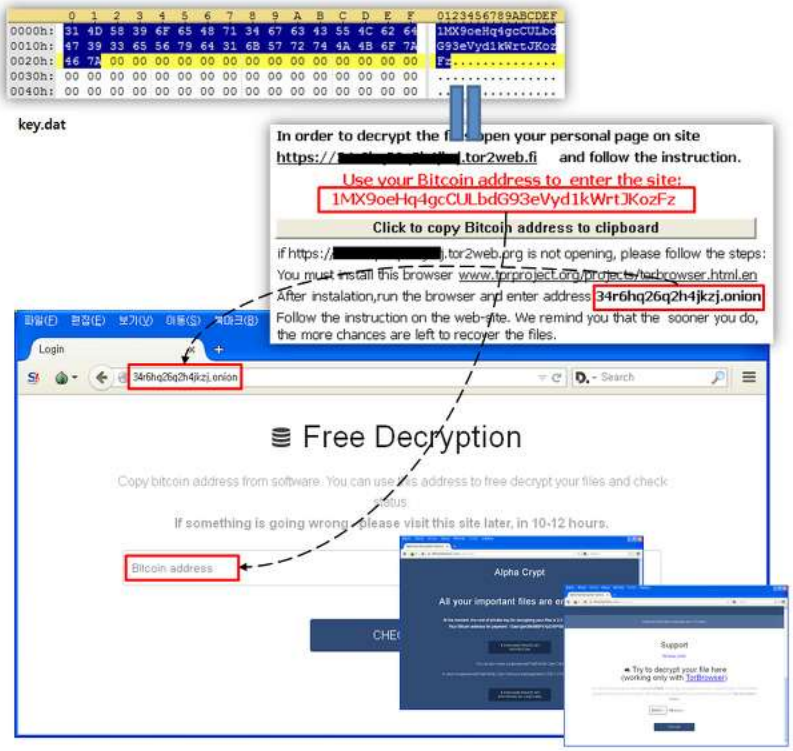
ここで「key=6446f0b0351eae14d01bcc70b0d4653ac1ce34887b2d23a53607c25655c8c793」の項目に追加される値（[図4]の①）がAESキーの値であり、「addr=1MX9oeHq4gcCULbdG93eVyd1kWrtJKozFz」項目に追加される値（[図4]の②）は、ビットコインアドレス（BitCoin Address）で使用される値である。

TeslaCryptランサムウェアに存在する攻撃者アドレスは、次の4箇所だった。

- TeslaCryptランサムウェアの攻撃者アドレス
  - 7tno4hib47vlep5o.s2.tor-gateways.de
  - 7tno4hib47vlep5o.tor2web.fi
  - 7tno4hib47vlep5o.tor2web.blutmagie.de
  - 7tno4hib47vlep5o.2kjb7.net

(2) ビットコインアドレス（Bitcoin Address）

[図5]のようにkey.datファイルの開始部分に存在する値「1MX9oeHq4gcCULbdG93eVyd1kWrtJKozFz」は、ファイル復元のため決済時に使用されるビットコインアドレスと同じ値である。正確なビットコインアドレス値を入力すると、プライベートキーを決済するための手段として、約 528 ドル相当のビットコイン（2.2 BTC～）またはペイパル（PayPal My Cash Card）など 2つの方法を提案する。



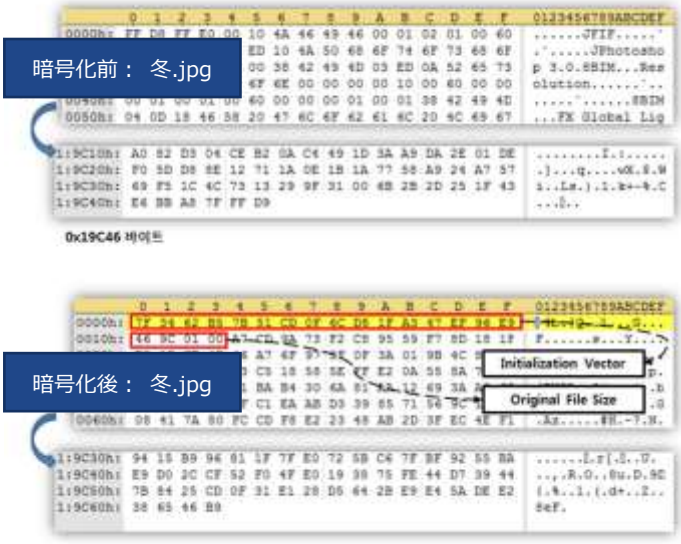
[図5] 決済を誘導するビットコインアドレス値

TeslaCryptランサムウェアはユーザーの信頼を得るため、試しに500kb以下の暗号化ファイルを 1個だけ復号化し、Webベースでファイルをアップロードする構造になっている。

2.感染ファイルの構造

[図6]は、TeslaCryptランサムウェアに感染された「冬.jpg」ファイルの感染前後を比較したものだ。感染後のファイルサイズが増加し、内部データも暗号化されたことが分かる。

TeslaCryptランサムウェアによって暗号化されたファイルは、「元のファイル名.ecc」の形式をしている。変更されたファイルの共通点は、ファイルの開始から0x10バイトのデータが、暗号化に使用された2次キー値 IVであり、後にLittle-endian形式で保存された4バイト値（0x00019C46）に、暗号化前の元ファイルのサイズ情報が保存されるという点である。つまり暗号化されたデータの開始は、感染ファイルの開始から0x14バイト離れた位置であることが分かる。



[図6] TeslaCryptランサムウェア感染前後のファイル比較

ファイルの復元作業は、次のような過程を経る。

① key.datファイルからAESキー情報0x20バイトを読み込む。（オフセット：0x177~0x197）

- 変換前のAESキー：6446f0b0351eae14d01bcc70b0d4653ac1ce34887b2d23a53607c25655c8c793

② AESキー値を「SHA256」値に変換する。

- 変換後のAESキー：795230585896ef6e3dfdbaf694f426e60290b36b00c5b75cf028a82f2a35afc3

③ 暗号化されたファイルから得たIV値とAESキーを使用して復号化する。この時、IV値は*.eccファイルの最初の16バイトである。

```
# read_key: 'key.dat'    から読み込んだ値
read_key = KeyBuff[0x177:0x197]
# SHA256 変換
h = SHA256.new()
h.update(read_key)
# aes_key:        復号する際に使用するAES Key
aes_key = h.digest()

# iv_key: '*.ecc'   から読み込んだ値
iv_key = EncBuff[:16]
# aes_key, iv_key    を利用して復号化
obj = AES.new(aes_key , AES.MODE_CBC, iv_key)
# enc_data: '*.ecc'    から読み込んだ値
enc_data  = encbuff[20:]
dec_data = obj.decrypt(enc_data)
```

[図7] ファイル復元のためのPythonコード

このような情報に基づいて、感染されたシステムの「key.dat」ファイルと暗号化されたファイル（*.ecc）があれば、[図7]のような Pythonコードで感染ファイルを復元できる。

3.その他key.datファイルの項目

key.datファイルに存在する8つのデータのうち、ビットコインアドレスとAESキーに当たる項目を中心に紹介した。key.datファイルに存在するその他の項目は、次のようなAPI呼出しによって得た情報から作成される。

[NETAPI32]

- NetStatisticsGet(0, "LanmanWorkstation", 0, 0, &buf)
- NetStatisticsGet(0, "LanmanServer", 0, 0, &buf)

[ADVAPI32]

- CryptAcquireContextW(&hProv, 0, 0, PROV_RSA_FULL, CRYPT_ VERIFYCONTEXT)
- CryptGenRandom(hProv, 0x40, &buf)
- CryptAcquireContextW(&hProv, 0, "Intel Hardware Cryptographic Seervice Provider", PROV_INTEL_SEC, 0)
- CryptGenRandom(hProv, 0x40, &buf)

[kernel32]

- CreateToolhelp32Snapshot(TH32CS_SNAPALL, 0)
- GetTickCount
- GetTickCount
- Heap32ListFirst
- Heap32First
- Heap32Next
- GetTickCount
- Heap32ListNext
- GetTickCount
- GetTickCount
- Process32First
- Process32Next
- GetTickCount
- GetTickCount
- Thread32First
- Thread32Next
- GetTickCount
- GetTickCount
- Module32First
- Module32Next
- GetTickCount
- GlobalMemoryStatus
- GetCurrentProcessId

[図8]は、上記API中「CryptGenRandom」関数を使用して得た0x40サイズのデータが、特定の関数（sub_41B3A0）によって変換される過程である。

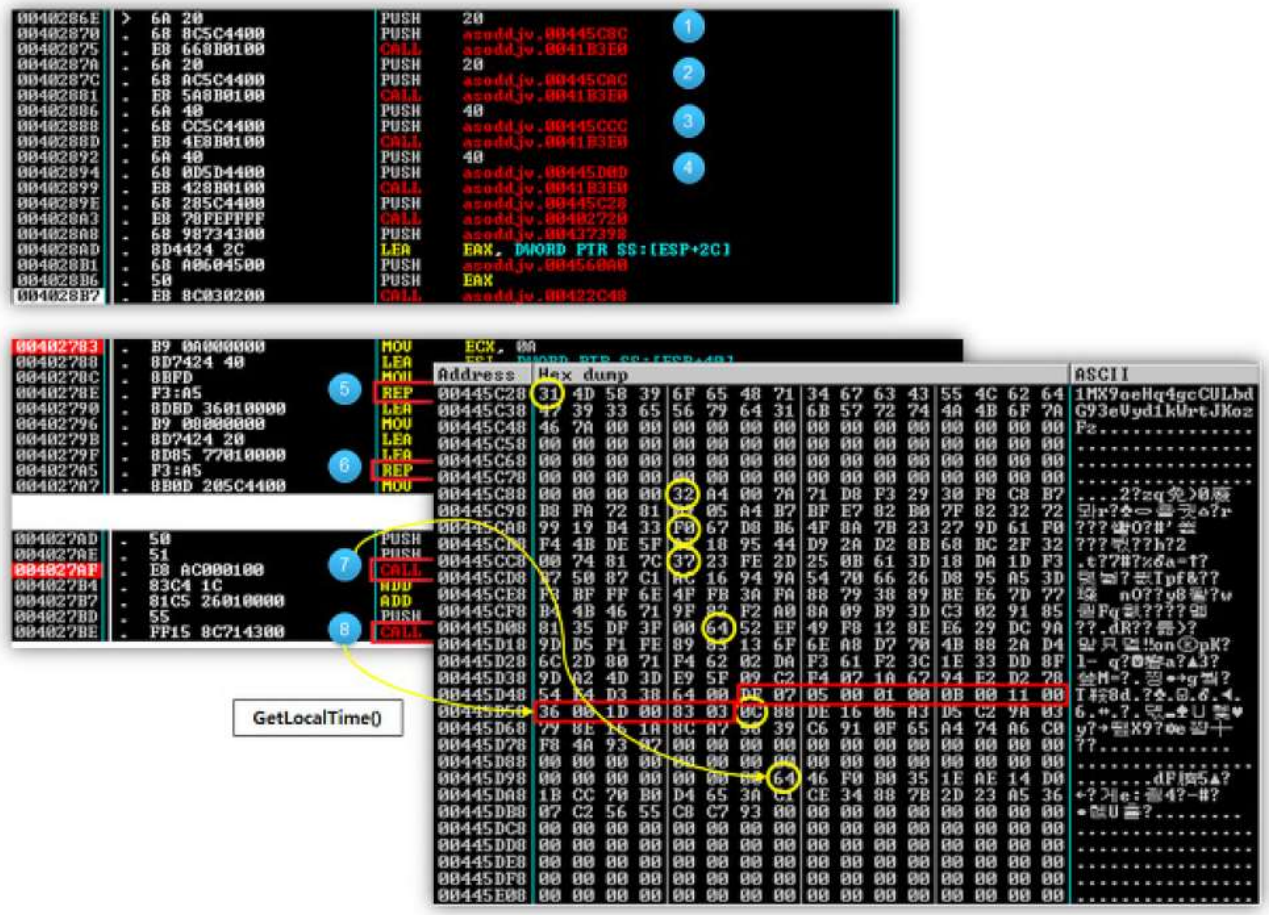
[図8] CryptGenRandom変換の過程

同様に、他のAPIの結果として出されたリターン値も、sub_41B3A0関数による変換処理を経て、最終変換値がkey.datに使用されることを確認できる。たとえばGlobalMemoryStatus、GetCurrentProcessId API呼出しで得られた結果値も、[図9]のようにsub_41B3A0関数による変換過程を経ている。

```
sub_41B410();
GlobalMemoryStatus(&Buffer);
sub_41B3A0((int)&Buffer, 0x20u, 1.0);
v26 = GetCurrentProcessId();
sub_41B3A0((int)&v26, '₩x04', 1.0);
return 1;
```

[図9] GlobalMemoryStatus、GetCurrentProcessId変換過程

[図10]は、key.datファイルの8つの詳細項目でデータを書込む（Write）過程を表している。赤いスクエアで表示された部分は、最後の8番目に追加される項目で、GetLocalTime APIを通じて得た時間情報（16バイト）である。（その他の各詳細な項目別データの意味と使用目的の分析は、今後提供予定）



[図10] key.datファイルの詳細項目

4.プロセス強制終了

感染スレッド（thread）起動後、2番目に作成されるスレッド（GetProcessImageFileNameW）は、右の名前で実行されるプロセスの強制終了（TerminateProcess）機能を持つ。単にプロセス名だけで比較されるため、名前を変更して実行すると通常の実行が可能だ。

● 強制終了されるプロセス

- taskmgr
- procexp
- regedit
- msconfig
- cmd.exe

ここまで、TeslaCryptランサムウェアの構造と動作を分析し、感染ファイルを復元する方法を紹介した。上記の方法でファイル復元が困難なユーザーのために、アンラボでは「.ecc」拡張子で暗号化されたファイル復元のため、専用のワクチンを提供している。ただし、key.datファイルが%AppData%パスに存在する場合のみ専用ワクチンを利用できる。

だがTeslaCrypt以外のほとんどのランサムウェアの場合、感染すると身代金を支払わない限りファイル復元の有効な手立てがない状況だ。

さらに、身代金を払ったとしてもシステムが元通りに復元されるという保証もないため、やはり事前にランサムウェアを遮断することが最善の方法といえよう。発信元が明確でない電子メールに添付された疑わしいファイルを実行しないこと。また、使用中のワクチンを常に最新の状態に更新し、OSやアプリケーションも最新バージョンに維持することが重要だ。そして重要ファイルは事前にバックアップをしておくことを推奨している。

V3製品群では、TeslaCryptランサムウェアを次のような診断名で検知している。

<V3診断名>

Trojan/Win32.Tescrypt

<参考サイト>

- https://technet.microsoft.com/en-us/library/cc788026.aspx
- https://pypi.python.org/pypi/pycrypto
- http://blogs.cisco.com/security/talos/teslacrypt
- https://blogs.mcafee.com/mcafee-labs/teslacrypt-joins-ransomware-field
- https://blog.kaspersky.com/teslacrypt-ransomware-targets-gamers/
- http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Ransom:Win32/Tescrypt.A#tab=2

http://jp.ahnlab.com

http://global.ahnlab.com

http://www.ahnlab.com

アンラボとは

株式会社アンラボは、業界をリードする情報セキュリティソリューションの開発会社です。
1995年から弊社では情報セキュリティ分野におけるイノベーターとして最先端技術と高品質のサービスをご提供できるように努力を傾けてまいりました。今後もお客様のビジネス継続性をお守りし、安心できるIT環境づくりに貢献しながらセキュリティ業界の先駆者になれるよう邁進してまいります。
アンラボはデスクトップおよびサーバー、携帯電話、オンライントランザクション、ネットワークアプライアンスなど多岐にわたる総合セキュリティ製品のラインナップを揃えております。どの製品も世界トップクラスのセキュリティレベルを誇り、グローバル向けコンサルタントサービスを含む包括的なセキュリティサービスをお届け致します。

AhnLab

〒105-0014 東京都港区芝4丁目13-2 田町フロントビル3階 | TEL: 03-6453-8315 (代)