Microsoft Storage Spaces SMP Service Overflow vulnerability

**environment**

Microsoft Windows Version 10.0.19041.423 64-bits

**Vulnerability description**

There is a memory overflow vulnerability in mispace.dll!ControlDispatch

**Causes of Vulnerability**

The ControlDispatch function will use the first 0x230 bytes of the rpc buffer to select different functions to call

I found vulnerabilities in these functions:

Spctl\_Subsystem\_SetAttributes\_L1,

Spctl\_ResiliencySetting\_SetDefaults\_P1,

Spctl\_Subsystem\_CreatePool\_L1,

Spctl\_Pool\_SetDefaults\_P1,

Spctl\_Pool\_CreateVolume\_P1,

Spctl\_StorageJob\_WriteMessage\_C1

These called functions have only one buffer parameter

These vulnerabilities are due to not verifying the value in the buffer

Vulnerability trigger path 1

Spctl\_StorageJob\_WriteMessage\_C1->CSpJobMgr::AddJobMessage->SmCopyString

The code that triggers the vulnerability in SmCopyString

if ( a1 )

{

if ( a2 )

{

v7 = -1i64;

do

++v7;

while ( a2[v7] ); //Access invalid memory

Vulnerability trigger path 2

Spctl\_Subsystem\_CreatePool\_L1->SuexGetDrives

The code that triggers the vulnerability in SuexGetDrives

while ( 1 )

{

v46 = 40;

v19 = (\_\_int128 \*)&v9[4 \* v18 + 1]; //Access invalid memory

//......

}

Vulnerability trigger path 3 and 4 and 5 and6 Spctl\_Pool\_CreateVolume\_P1->GetCreateVolumeParams

Spctl\_Pool\_SetDefaults\_P1->GetPoolDefaults

Spctl\_ResiliencySetting\_SetDefaults\_P1->GetResiliencySettingDefaults

Spctl\_Subsystem\_SetAttributes\_L1->GetSubsystemAttributes

The vulnerability points of these functions are the same :

if ( a2 >= 0x40 )

{

v7 = a1[3];

v8 = LocalAlloc(0x40u, v7);

v9 = v8;

if ( v8 )

{

memcpy\_0(v8, a1, v7); // Unverified length

\*a3 = v9;

}

**POC**

The successful execution of poc requires the Microsoft Storage Spaces SMP service to be started first, which is a necessary condition You can enter from cmd

sc config smphost start = auto

sc start smphost

Then execute poc.exe, the Microsoft Storage Spaces SMP service will crash