Remote Access API BUG

**environment**

Microsoft Windows Version 10.0.19042.1165 64-bits

**Vulnerability description**

RASAPI32!DwRemoveEapUserInfo

Memcpy length integer overflow in DwRemoveEapUserInfo causes heap overflow

**Causes of Vulnerability**

The api call where the vulnerability is located is RasSetEapUserDataW. When this function is called, ReadEntryFromSystem is first called to ensure whether the pbk file exists and whether the entry is valid. When the check is passed, DwSetEapUserInfo will be created and opened. HKEY\_USERS\current user SID\SOFTWARE\Microsoft\RAS EAP\ UserEapInfo registry key, and create and set the value of EapInfo

If the first 4 bytes of this item are 0x31504145, DwRemoveEapUserInfo, which is the vulnerable function, will be called. The problem here is that the value of EapInfo is completely controllable, and memmove\_0 will be counted based on the value offset of 0x18 when calling memmove\_0 Operate to get the final length, and there is no effect

if (\*a3)

{

do

{

if (\*a1 == \*((\_QWORD \*)v12 + 1) && a1[1] == \*((\_QWORD \*)v12 + 2) && a6 == \*((\_DWORD \*)v12 + 1))

break;

v11 += (\*((\_DWORD \*)v12 + 6) + 39) & 0xFFFFFFF8;

v12 = &a2[v11];

}

while (v11 <v6 );

if (v11 <v6)

{

v13 = (\*((unsigned int \*)v12 + 6) + 39i64) & 0xFFFFFFFFFFFFFFF8ui64;

cbData = v6-((\*((\_DWORD \*)v12 + 6) + 39) & 0xFFFFFFF8);

memmove\_0((void \*)&a2[v11], &a2[v13 + v11], v6-v11-v13);

if (a5)

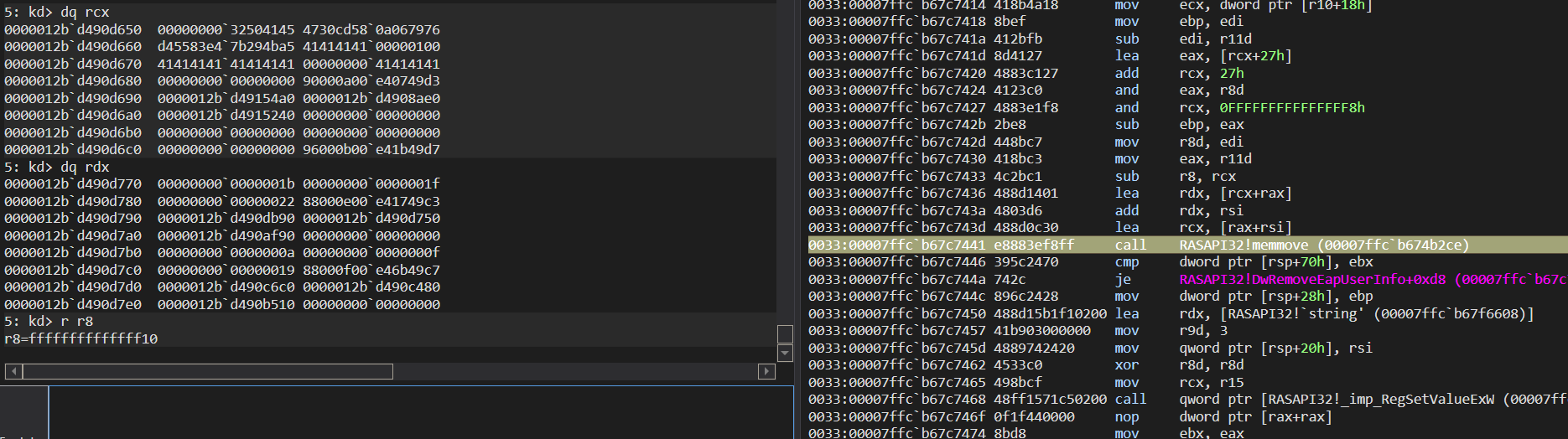
v7 = RegSetValueExW(a4, L"EapInfo", 0, 3u, lpData, cbData);

else

\*a3 = cbData;

}

}



**POC**

I made a Poc to verify that there is a problem with this API

When poc is executed, it will first call RasSetEapUserDataW to initialize the registry, then set EapInfo, and finally set to call RasSetEapUserDataW to trigger an overflow.

Put 1.pbk and poc.exe in the same directory, execute them, poc will stop working

Expected print result

