

Re: HOWTO Validate security privileges

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.platformsdk.security/2005-12/msg00142.html>

- *From:* ATS <ATS@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 20 Dec 2005 07:03:02 -0800
-

Thanks for the reply Alex,

I'm having a problem though. I'm trying to verify that the currently logged in user/process has SE_TCB_NAME otherwise known as "Act as part of the operating system". My logon Id has this right, by being in the "local" Administrators group, which on my machine (that is running in its own domain), I have explicitly set for all local Administrators to have "Act as part of the operating system". But when I run GetTokenInformation, and check the list of returned privileges, none of them have the LUID for SE_TCB_NAME, that I got from "LookupPrivilegeValue(NULL, SE_TCB_NAME, &LUID_Temp)".

Here is my logic:

```
#define AFX_THROW AfxThrowUserException()

BOOL bRet;
BYTE *pBYTE_Privileges = NULL;
HANDLE HANDLE_CurrentUser = NULL;
CString
csSecurityRequirements,
ERR,
LA;

try
{
LA = "initialize security parameters";

BOOL bHasPrivileges = FALSE;
LUID_AND_ATTRIBUTES *pLUID_AND_ATTRIBUTES = NULL;

CMapStringToPtr cmstpPrivileges;
LUID zLUID[100];

LUID *pLUID;
int iLUID = 0;

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_TCB_NAME, pLUID);
```

Re: HOWTO Validate security privileges

```
cmstpPrivileges.SetAt("Act as part of the operating system", (void *)
pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_CREATE_TOKEN_NAME, pLUID);
cmstpPrivileges.SetAt("Create a token object", (void *) pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_CREATE_PERMANENT_NAME, pLUID);
cmstpPrivileges.SetAt("Create permanent shared objects", (void *) pLUID);

// The MSDN documentation says SE_IMPERSONATE_NAME exists, but I can not
// find a header file
// for it.
//
// pLUID = &zLUID[iLUID++];
// bRet = LookupPrivilegeValue(NULL, SE_IMPERSONATE_NAME, pLUID);
// cmstpPrivileges.SetAt("Impersonate a client after authentication",
// (void *) pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_BATCH_LOGON_NAME, pLUID);
cmstpPrivileges.SetAt("Log on as a batch job", (void *) pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_SERVICE_LOGON_NAME, pLUID);
cmstpPrivileges.SetAt("Log on as a service", (void *) pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_INTERACTIVE_LOGON_NAME, pLUID);
cmstpPrivileges.SetAt("Log on locally", (void *) pLUID);

pLUID = &zLUID[iLUID++];
bRet = LookupPrivilegeValue(NULL, SE_TAKE_OWNERSHIP_NAME, pLUID);
cmstpPrivileges.SetAt("Take ownership of files or other objects", (void *)
pLUID);

LA = "open the current process's security token";

bRet = OpenProcessToken
(
GetCurrentProcess(), TOKEN_QUERY, &HANDLE_CurrentUser
);

if (!bRet)
{
ERR.Format("Error %08X.", GetLastError());
ERR += csSecurityRequirements;
AFX_THROW;
}
```

Re: HOWTO Validate security privileges

```
LA = "examine security for the current user's privileges";

CString csPrivilege;
DWORD dwTemp = 0, dw;
POSITION POSITION_Temp;
TOKEN_PRIVILEGES *pTOKEN_PRIVILEGES;
pBYTE_Privileges = new BYTE[sizeof(TOKEN_PRIVILEGES)];
pTOKEN_PRIVILEGES = (TOKEN_PRIVILEGES *) pBYTE_Privileges;

GetTokenInformation
(
HANDLE_CurrentUser,
TokenPrivileges,
(void *) pTOKEN_PRIVILEGES,
sizeof(TOKEN_PRIVILEGES),
&dwTemp
);

delete [] pBYTE_Privileges;
pBYTE_Privileges = new BYTE[dwTemp + 1000];
pTOKEN_PRIVILEGES = (TOKEN_PRIVILEGES *) CBlob_Privileges.pBYTE;

GetTokenInformation
(
HANDLE_CurrentUser,
TokenPrivileges,
(void *) pTOKEN_PRIVILEGES,
(DWORD) CBlob_Privileges.uiSize,
&dwTemp
);

if (!bRet)
{
ERR.Format("Error %08X.", GetLastError());
ERR += csSecurityRequirements;
AFX_THROW;
}

for (POSITION_Temp = cmstpPrivileges.GetStartPosition(); POSITION_Temp;)
{
cmstpPrivileges.GetNextAssoc(POSITION_Temp, csPrivilege, (void *)& pLUID);
bHasPrivileges = FALSE;

for (dw = 0; dw < pTOKEN_PRIVILEGES->PrivilegeCount; dw++)
{
pLUID_AND_ATTRIBUTES = &pTOKEN_PRIVILEGES->Privileges[dw];

if
(
(pLUID_AND_ATTRIBUTES->Luid.HighPart == pLUID->HighPart)
&&
```

Re: HOWTO Validate security privileges

```
(pLUID_AND_ATTRIBUTES->Luid.LowPart == pLUID->LowPart)
)
{
if (pLUID_AND_ATTRIBUTES->Attributes == SE_PRIVILEGE_ENABLED)
{
bHasPrivileges = TRUE;
break;
}
}
}

if (!bHasPrivileges)
{
ERR.Format
(
"Your account does not have all required security privileges. It is
missing "
"security permission to \"%s\". For reference, your account must
have these "
"security privileges:\n"
"\n"
" 'Act as part of the operating system'\n"
" 'Create a token object'\n"
" 'Create permanent shared objects'\n"
" 'Impersonate a client after authentication'\n"
" 'Log on as a batch job'\n"
" 'Log on as a service'\n"
" 'Log on locally'\n"
" 'Take ownership of files or other objects'\n"
"\n",
csPrivilege
);
AFX_THROW;
}
}

}
catch(...)
{
}

if (pBYTE_Privileges)
{
delete [] pBYTE_Privileges;
}

if (HANDLE_CurrentUser)
{
CloseHandle(HANDLE_CurrentUser);
HANDLE_CurrentUser = NULL;
}
```

- **References:**

- ◆ **Re: HOWTO Validate security privileges**

- ◇ *From:* Alex Fedotov

- Prev by Date: **Re: [GINAHOOK] Pass logon data to Winlogon**

- Next by Date: **Information required about "how to elevate rights" and "how to install service with restricted access on windows"**

- Previous by thread: **Re: HOWTO Validate security privileges**

- Next by thread: **Capicom: transform PKCS12 into PKCS8 for signature ?**

- Index(es):

- ◆ **Date**

- ◆ **Thread**