Laboratorul 6

Security Descriptor

Orice obiect din sistemul de operare Windows are asociat un security descriptor. Security descriptor-ul reprezinta o structura care contine informatii despre modalitatea in care un utilizator poate accesa obiectul.

Componentele security descriptorului sunt urmatoarele:

1. Header (care contine de fapt niste informatii legate de componenentele prezente in structura si o serie de flaguri specifice fiecarui tip de obiect in parte)
2. Owner
3. Group
4. DACL
5. SACL

1. Header:

   Header-ul contine informatii despre locatia urmataorelor componente (relativ sau absolut) precum si o serie de flaguri. Un layout relativ inseamna ca fiecare componente a security descriptor-ului va porni de la un offset fata de header (fisierele folosesc asa ceva), pe cand la o schema absoluta, security descriptor-ul va contine pointer catre fieacre componenta, acestea putand fi pozitionate oriunde in memorie. Flagurile cele mai importante se refera la faptul daca sunt sau nu prezente componentele DACL si SACL precum, daca acestea trebuie mostenite de la parinte (de ex, un fisier sa aiba aceleasi dreturi ca directorul din care face parte), precum si daca acestea au valori standard.
2. Owner:

   Contine un SID ce identifica proprietarul obiectului.
   Un SID (security identifier) reprezinta un numar cu o anumita structura prin care orice utilizator sau grup de pe sistem este identificat.
   Structura unui SID este urmatoarea:

   - Revision number (care deocamdata este tot timpul 1)
   - Un numar ce identifica autoritatea si subautoritatea ce a creat acel SID
   - Un numar relativ, numit RID, unic pentru autoritatea ce a emis acel SID

   Pentru a vedea o lista de SID-uri pentru sistemul vostru, puteti sa va uitati la cheile din registry de sub hive-ul (HKEY_USERS).

   Un exemplu de sid este urmatorul:

   S-1-5-21-1463437245-1224812800-863842198-1128

Sid-ul incepe deobicei cu litera S, ca sa indice faptul ca lucram cu un SID, urmeaza apoi valoarea 1 ce reprezinta revizia (Revision Number). Urmatoarea componenta (5), reprezinta autoritatea care a creat sid-ul. Acesta poate lua o valoarea de la 0 la 5. Cea mai intalnita valoarea este 5 (SECURITY_NT_AUTHORITY), specifica autoritatii ce creaza majoritatea user-ilor si grupurilor de pe sistem.

Umatoarea componenta, 21-1463437245-1224812800-863842198, id-urile de subautoritati. Sistemul de operare creaza 3 subautoritati in mod aleatoriu pentru a se asigura ca sid-ul este distinct (21 reprezinta subclasa ce anunta ca dupa el urmeaza o subautoritatea.) Acest SID poate fi specific unui domeniu, in caz ca user-ul apartine unui domeniu, sau poate identifica un sistem local.

Ultima componenta, numita RID (aici cu valoarea de 1128) porneste de la 1000 pentru utilizatorii si grupurile din sistem si este incrementata cu 1 de fiecare data cand un utilziator este adaugat. Valorile pana la 1000 sunt rezervate pentru sistem. (De ex: 500 reprezinta RID-ul pentru contul de Administrator, iar 501 pentru Guest)

Nu toate sid-urile sunt unice. Sunt unele sid-uri, numite well known seeds, care sunt la fel pentru orice sistem si au rolul de grupuri generice (ex. Grupul Everyone) sau rolul de place-holdere. (Ex. Seed-ul pentru creatorul fisierului)

3. SID-ul grupului din care face parte user-ul ce detine obiectul
4. Informatii despre DACL si SACL

   Informatiile despre cine are si cine nu are permisiunea de a face o actiune pe un obiect se afla intr-o tabela numita Discretionay Access Control List (DACL).
   Aceastra tabela, reprezinta de fapt un vector de Access Control Entries (ACEs)
   Un Access Control Entry reprezinta o structura ce specifica pentru un singur SID, daca ii este permis sau nu un drept. Deci, un ACE este format din urmatoarele campuri:

   - SID
   - Drept
   - Un Flag ce semnifica daca se permite sau nu dreptul specificat mai sus

   Aceasta lista de ace-uri, numita ACL, este sortata astfel incat intrarile care blocheaza accesul sa fie primele. Acest lucru este important deoarece exista posibilitatea ca pentru un user sa avem interdictie pe un drept, dar grupul din care apartine sa aiba permis acelasi drept. Daca ACE-ul specific grupului (cu permisiunea allow) ar fi in fata celui specific utilizatorului (cu permisiunea deny), atunci utilizatorul ar primi accesul pentru obiectul respectiv.

   Daca pentru un SID sau grupul din care face nu are un ACE intr-o tabela DACL, atunci se considera ca acel SID are deny pe toate drepturile. Din acelasi motiv, daca un DACL este gol (nu are nici un ACE) atunci nici un utilizator nu are nici un drept asupra obiectului. Doar proprietarul poate adauga drepturi noi.

Un caz de exceptie este atunci cand un security descriptor nu are nici un DACL. (faceti diferenta intre cazul cand DACL gol si cand lipseste). In cazul acesta, oricine are toate drepturile asupra fisierului respectiv.

Tabela SACL are aceeasi structura ca si tabela DACL, diferenta o constituie felul in care este folosita. Tabela SACL este utila atunci cand un administrator de sistem doreste sa vada cand un utilizator a reusit sa acceseze (sau nu a putut sa acceseze) un obiect cu un anumit drept. Aceste informatii vor fi scrise intr-un event log.

Crearea de security descriptor

Pentru a crea un security descriptor, trebuie intai alocata memorie, iar apoi initializata structura. Pentru a aloca memorie se foloseste o functie standard de alocare (malloc, new sau LocalAlloc/GlobalAlloc), iar numarul de bytes ce trebuie alocati este SECURITY_DESCRIPTOR_MIN_LENGTH.

Aceasta zona de memorie va fi trimisa mai departe pentru initializarea unui security descriptor. Acest lucru se face cu ajutorul functiei InitializeSecurityDescriptor.

```
BOOL WINAPI InitializeSecurityDescriptor(
  _Out_  PSECURITY_DESCRIPTOR pSecurityDescriptor,
  _In_   DWORD dwRevision
);
```

pSecurityDescriptor	Un buffer de minim SECURITY_DESCRIPTOR_MIN_LENGTH bytes care va tine noul security descriptor
dwRevision	SECURITY_DESCRIPTOR_REVISION

Functia creaza un in zona de memorie data ca parametru un security descriptor gol, in format absolut care contine doar revision number (fara owner, grup, SACL sau DACL).

Mai departe, in functie de necesitate, trebuie adaugate restul structurilor.

- **Adaugarea owner-ului si al grupului**

Owner-ul este reprezentat de o structura de tip SID. Acesta ar trebui in mod normal obtinut sau creat. Pentru detalii despre cum se face asta, vezi mai jos Creare/Obtinere SID. Odata obtinut, se apeleaza functia SetSecurityDescriptorOwner.

```
BOOL WINAPI SetSecurityDescriptorOwner(
  _Inout_   PSECURITY_DESCRIPTOR pSecurityDescriptor,
  _In_opt_  PSID pOwner,
  _In_      BOOL bOwnerDefaulted
);
```

pSecurityDescriptor	Pointer la security descriptor-ul la care se doreste sa fie adaugat/modifcat owner-ul
pOwner	Pointer la structura de tip SID ce identifica un user. Daca este setat pe NULL, se elimina owner-ul din structura sid-ului
bOwnerDefaulted	Aproapte tot timpul false. Indica daca este un owner default

Intr-un mod asemanator se adauga si grupul prin functia SetSecurityDescriptorGroup care are aceiasi parametri. Diferenta o consta in faptul ca in loc de SID-ul unui user, functia va primi SID-ul unui grup.

- **Adaugarea/Modificarea DACL-ului si SACL-ului**

Atat modificarea cat si crearea unei tabele ACL se face cu ajutorul functiei SetEntriesInACL. Aceasta functie primeste o lista de access controll entries (ACEs) si optional, o lista veche ACL, iar in ultimul parametru se va obtine noua tabela de ACL .

```
DWORD WINAPI SetEntriesInAcl(
  _In_      ULONG cCountOfExplicitEntries,
  _In_opt_  PEXPLICIT_ACCESS pListOfExplicitEntries,
  _In_opt_  PACL OldAcl,
  _Out_     PACL *NewAcl
);
```

cCountOfExplicitEntries	Numarul de ACEs
pListOfExplicitEntries	Vectorul de ACEs
OldAcl	O lista existenta de ACL-uri. Daca acest parametru este NULL, atunci va fi creata o noua lista. Altfel, va fi create o lista pornind de la cea existent
NewAcl	Functia va scrie aici un pointer catre noua tabela ACL. Dupa ce este folosit, aceastra structura trebuie eliberata

In caz de succes, functia intoarce ERROR_SUCCESS. Altfel, intoarce un cod de eroare.

Structura din Windows API specifica unui ACE este EXPLICIT_ACCESS.

```
typedef struct _EXPLICIT_ACCESS {
  DWORD       grfAccessPermissions;
  ACCESS_MODE grfAccessMode;
  DWORD       grfInheritance;
  TRUSTEE     Trustee;
} EXPLICIT_ACCESS,
```

grfAccessPermissions	Permisiunile la care se refera acest ACE. Deobicei, in acest camp sunt puse permisiunile specifice fiecarui obiect, de ex:KEY_READ pentru o cheie de registry sau PROCESS_TERMINATE pentru un proces
grfAccessMode	Specifica daca drepturile de mai sus vor fi setate sau blocate. Cele mai uzuale valori sunt SET_ACCESS si DENY_ACCESS. Pentru SACL, pot fi folosite constantele SET_AUDIT_SUCCESS si SET_AUDIT_FAILURE
grfInheritance	Specifica cum va fi mostenit acest ACE. Valori uzuale sunt NO_INHERITANCE sau SUB_CONTAINERS_AND_OBJECTS_INHERIT
Trustee	Reprezinta o structura ce identifica user-ul sau grupul. Din aceasta structura, trebuie setati urmatorii membrii: • Trustee.TRUSTEE_FORM: specifica daca userul este identificat prin nume sau SID. Valori uzuale sunt TRUSTEE_IS_SID si TRUSTEE_IS_NAME • Trustee.TrusteeType: identifica tipul de SID (user, grup, etc.). Valori uzuale sunt: TRUSTEE_IS_USER, TRUSTEE_IS_GROUP, TRUSTEE_IS_WELL_KNOWN_GROUP. • Trustee.ptstrName: numele sau structura sub forma de PSID ce identifica un cont. Numele sau SID-ul for fi alese in functie de TRUSTEE_FORM

Setarea unei liste ACL la un security descriptor se face cu ajutrul functiilor SetSecurityDescriptorDacl/ SetSecurityDescriptorSACL

Dupa cum observati, exista o functie care seteaza si intoarce orice membru din structura security descriptor. Totui, exista si 2 functii generice,care pot fi folosite fie pentru a adauga oricare din membrii structurii security descriptor sau chiar pe toti deodata.

Acestea sunt: GetNamedSecurityInfo si SetNamedSecurityInfo

- **Creare/Obtinere SID**

  Daca se doreste obtinerea SID-ului unui anumit user, se poate apela functia LookupAccountName, ce primeste numele unui uesr si intoarce SID-ul. (Exista si functia LookuAccountSid, care face actiunea inversa)

  BOOL WINAPI LookupAccountName(

```
_In_opt_   LPCTSTR lpSystemName,
_In_       LPCTSTR lpAccountName,
_Out_opt_  PSID Sid,
_Inout_    LPDWORD cbSid,
_Out_opt_  LPTSTR ReferencedDomainName,
_Inout_    LPDWORD cchReferencedDomainName,
```

```
  _Out_      PSID_NAME_USE peUse
);
```

lpSystemName	Numele sistemului pe care sa se caute userul. Daca se speciica NULL, atunci se va face cautarea pe sistemul curent, iar apoi pe domeniul din care face parte sistemul
lpAccountName	Numele user-ului
Sid	Un pointer catre un buffer. Aici va fi pus sid-ul userului dup ace este gasit
cbSid	Dimensiunea in bytes a buffer-ului Sid. Daca functia esueaza pentru ca buffer-ul SID nu a fost sufficient, aici se vor scrie numarul de bytes necesari pentru a tine SID-ul.
ReferencedDomainName	Un buffer ce primeste numele sistemului/domeniului pe care se afla user-ul
cchReferencedDomainName	Dimensiunea in bytes a ReferencedDomainName. Daca numele domeniului nu incape in buffer-ul specificat, atunci aici va fi pus dimensiunea necesara a buffer-ului.
peUse	Pointer catre o variabila ce primeste tipul de cont. (vf enumerarea SID_NAME_USE)

Totusi, in unele cazuri, este necesara initializarea unei structuri de tip SID. Un astfel de exemplu ar fi atunci cand vrem sa adaugam drepturi specifice pentru Owner-ul unui fisier. Intrucat owner-ul se poate schimba, nu are sens sa adaugam un sid specific unui nume. Pentru aceste cazuri, exista SID-uri numite Well Known SIDs.

Well Known Sids reprezinta un grup de sid-uri pe care orice sistem Windows il recunoaste (de ex. Grupul de administrator, grupul Everyone, contul de administrator, etc.), precum si o serie de SID-uri pe post de sablon. Acestea din urma vor fi inlocuite la verificare cu SID-ul corespunzator. (de ex, sid-ul care identifica cine a creat obiectul, va fi inlocuit la verificare cu sid-ul user-ului corespunzator).

O functie utila pentru a initializa un astfel de sid, este CreateWellKnownSid

```
BOOL WINAPI CreateWellKnownSid(
  _In_       WELL_KNOWN_SID_TYPE WellKnownSidType,
  _In_opt_   PSID DomainSid,
  _Out_opt_  PSID pSid,
  _Inout_    DWORD *cbSid
);
```

WellKnownSidType	Reprezinta o enumerare de sid-uri well-known. In functie de necesitate, se alege sid-ul corespunzator: https://msdn.microsoft.com/en-us/library/windows/desktop/aa379650(v=vs.85).aspx
DomainSid	Indica domeniul la care se refera sid-ul. Daca se doreste sistemul local, atunci se trimite NULL
pSid	Un buffer unde va fi memorat sid-ul. De preferat sa fie alocat de dimensiunea SECURITY_MAX_SID_SIZE

cbSid	Numarul de bytes alocati pentru buffer-ul pSid. Variabila va fi suprascria cu numarul de bytes scisi in buffer.

Utilizarea unui security descriptor:

In general, un security descriptor este atasat unui obiect, la creare, prin intermediul unei structuri de tip SECURITY_ATTRIBUTES

```
typedef struct _SECURITY_ATTRIBUTES {
  DWORD  nLength;
  LPVOID lpSecurityDescriptor;
  BOOL   bInheritHandle;
} SECURITY_ATTRIBUTES, *PSECURITY_ATTRIBUTES, *LPSECURITY_ATTRIBUTES;
```

nLength	Dimensiunea structurii, intotdeauna va fi sizeof (SECURITY_ATTRIBUTES);
lpSecurityDescriptor	Un pointer catre structura security descriptor
bInheritHandle	Specifica daca handle-ul va fi mostenit sau nu

Alte functii utile pentru security descriptor:
EqualSid,IsWellKnownSid, AllocateAndInitializeSid