Sistemas Operativos 2

2021/22

Segurança no modelo de programação Windows

Conceitos e API

1

Tópicos

Conceitos de segurança
Privilégios e direitos
Acesso discricionário
API
Exemplos

Bibliografia específica para este capítulo:

- Advanced Windows (3rd Edition); Jeffrey Richter
- WindowsNT 4 Programming; Herbert Schildt

- MSDN Library – PlattformSDK: DLLs, Processes, and Threads (disponível online e no ISEC)

2

Segurança em WindowsNT/Win32

Conceitos principais

- **Direitos**
  - Capacidade dada a utilizadores ou grupos de agir de uma determinada forma **sobre determinados objectos**
  - Exemplos
    - Leitura em determinado ficheiro
  - São associados ao **objecto** em questão
- **Privilégios**
  - Capacidades predefinidas de utilizadores ou grupos efectuarem determinadas **operações** no sistema
  - Exemplos:
    - Instalar device drivers
    - Efectuar backups
  - São associados ao **utilizador** ou grupo

Segurança em WindowsNT/Win32

Exemplos de direitos para ficheiros

- Leitura, escrita, execução, eliminação, alteração dos atributos, alteração da propriedade

Exemplos de privilégios

- Modificar o relógio do sistema
- Aumentar a prioridade de uma *thread*
- Obter a propriedade (posse) de um objecto
- Desligar o sistema
- Correr programas em modo *debug*

Identificados por constantes

Exemplo

- ACCESS_SYSTEM_SECURITY – Privilégio de modificar aspectos de segurança

Segurança em WindowsNT/Win32

Objectos a cujo acesso é controlado pelo sistema

- Ficheiros e directorias
- Chaves no registry
- Dispositivos
- Pipes anónimos e com nome
- Processos e threads
- Rescursos partilhados em rede
- Objectos de sicronização
- Etc.

Segurança em WindowsNT/Win32

Estruturas principais de gestão de direitos e privilégios

- **Tokens** de acesso (tokens)
- **Descritores de segurança** (**SD** – Security Descriptor)
- **Identificadores de segurança** (**SID** – Security ID)
- **Listas de controlo de acesso** (**ACL** – Access Control List)
  - De sistema – SACL
  - **Discrecionárias DACL**

Gestão no núcleo ("*Executive NT*")

- O *Security Reference Monitor* (SRM) é o módulo que compara a ACL do objecto acedido com a informação do token, e valida o acesso pretendido.
- O SRM também é responsável pela auditoria (compara o token com o SACL)

- Local Security Authority (LSA): módulo responsável pela gestão de tokens

Segurança em WindowsNT/Win32

Tokens de acesso

- Associados aos processos
  (Indirectamente ao utilizador – em nome do qual corre esse processo)
- Identificam os privilégios do utilizador em nome do qual o processo corre

Conteúdo:
- SID do utilizador
- SIDs dos grupos a que o utilizador pertence,
- Privilégios atribuídos ao utilizador,
- ACL default

Segurança em WindowsNT/Win32

Tokens de acesso

- O utilizador obtém um token após o login bem sucedido no sistema
- Todos os processos lançados pelo utilizador correm com o seu token

O LSA (Local Security Authority) é o módulo responsável pela validação e geração dos tokens.

É possivel lançar um processo com identificação de outro utilizador (*user impersonation*) desde que o utilizador inicial possua os privilégios necessários para essa acção

Segurança em WindowsNT/Win32

Listas de acesso

- Descrevem o que é que cada utilizador/grupo pode / não pode fazer com determinado objecto
- Contém: 0, 1, ou mais entradas **ACE** (ACE = Access Control Entry)
- Uma ACE é um tuplo que descreve: utilizador-acesso-permite/não permite
- Os acessos negados (não permite) têm precedência sobre os permitidos

Variantes:

- **DACL**: (*discretionary*): descrevem os direitos de acesso ao objecto por utilizadores (definidos pelo dono do objecto)
- SACL: (de sistema) descreve (identifica) as acções de auditoria a efectuar ao acessos a um objecto

ACL nula (não existente): os acessos são todos permitidos

ACL vazia (existe e é vazia): nada é permitido

As *não-permissões* sobrepõem-se às *permissões ("o não vence o sim")*

Segurança em WindowsNT/Win32

Descritores de segurança (SD)

- Determinam os direitos de acesso ao objecto
- Estão associados a objectos
- Os direitos são descritos por listas de acesso (ACL)
- Aparecem em duas formas
  - Self-relative
    - Estruturas dinâmicas baseadas em ponteiros
    - Mais fáceis de modificar
  - Absolute
    - Estruturas autocontidas
    - Mais fáceis de guardar em ficheiro

API para converter entre os dois formatos

- MakeAbsoluteSD
- MakeSelfRelativeSD

Segurança em WindowsNT/Win32

Descritores de segurança – associado a um objecto

Conteúdo

- **SID do dono** (do objecto em questão)
- SID do grupo primário dono do objecto
- **DACL** – Controla o acesso de utilizadores ao objecto em questão
- **SACL** (ACL de sistema) – Determina que acessos vão ser auditados

Segurança em WindowsNT/Win32

Identificadores de segurança (SID)

- Identificador de utilizador ou grupo único a nível de sistema
  Exemplo: S-1-5-21-2313843232-6739283392-1020
- Tem uma estrutura interna constituída por campos:
  - Revision Number
  - Identifier Authority
    0 - Null Authority, 1 - World Authority, 2 - Local Authority
    3 - Creator Authority, 4 - Non-unique Authority, 5 - NT Authority
    9 - Resource Manager Authority
  - Identificador da máquina (local ou domínio) - 48 bits
    Exemplo: 21-3623811015-3361044348-30300820
  - Relative Identifier (RID) - 32 bits
    - (user ou grupo criados (não-default) têm um valor >= 1000)
- Armazenados no Security Account Database (HKEY_LOCAL_MACHINE\SAM\SAM) com acesso restrito inclusive aos administradores

Segurança em WindowsNT/Win32

API - Tokens (exemplos)

AdjustTokenGroups	Modifica os grupos num token
AdjustTokenPriviledges	Modifica privilégio num token
OpenThreadToken	Obtém o token de uma thread
OpenProcessToken	Obtém o token de um processo
CreateProcessAsUser	CreateProcess com um determinado Token
ImpersonateLoggedOnUser	Permite a uma thread personificar (através do token) um utilizador já logado
SetThreadToken	Atribui um token novo a uma thread (para impersonation)
GetTokenInformation	Obtém user, group, priviégios etc.
SetTokenInformation	Modifica o token

Segurança em WindowsNT/Win32

API - SD / SID (exemplos)

MakeAbsoluteSD	Obtém um SD absolute a partir de um self-relative
MakeSelfRelativeSD	Obtém um SD self-relative a partir de um SD absolute
InitializeSecurityDescriptor	Inicializa um novo SD (nenhum direito concedido a ninguém)
GetSecurityDescripterowner	Obtém o SID o dono do SD
GetSecurityDescriptorDacl	Obtém o ponteiro para a DACL do SD
AllocateLocallyUniqueId	Cria um SID único
LookupAccountSID	Obtém o nome da conte e domínio de um SID

Segurança em WindowsNT/Win32

API - ACL / ACE (exemplos)

InitializeAcl	Cria uma estrutura ACL nova
GetAclInformation	Obtém os dados (ex: tamanho, num ACEs)
SetAclInformation	Modifica uma (ex: numero de revisão)
AddAce	Acrescenta uma ou mais ACEs
AddAuditAccessAce	Acrescenta uma ACE de auditoria (à SACL)
GetSecurityDescriptorDacl	Obtém o ponteiro para a DACL do SD
DeleteAce	Apaga o n-ésima ACE
GetAce	Obtém o n-ésima ACE

Segurança em WindowsNT/Win32

API - Privilégios (exemplos)

Os privilégio são descritos por LUID – *Locally Unique Identifier* – que podem variar de máquina para máquina

LookUpPrivilegeName	Obtém o nome de um privilégio
LookUpPrivilegeValue	Obtém o identificador de um privilégio

Segurança em WindowsNT/Win32

API - Consulta/atribuição segurança (exemplos)

GetFileSecurity	Obtém um SD associado a um ficheiro/directoria
SetFileSecurity	Actualiza a descrição de segurança de um ficheiro/directoria através do SD indicado
GetUserObjectSecurity	*Idem* para objectos de utilizador
SetUserObjectSecurity	
GetKernelObjectSecurity	*Idem* para objectos do sistema
SetKernelObjectSecurity	

Segurança em WindowsNT/Win32

Estratégia genérica para executar operações privilegiadas

1. Invocar **OpenProcessToken()** com pelo menos as seguintes flags TOKEN_ADJUST_PRIVILEGE e TOKEN_QUERY
2. Utilizar **LookupPrivilegeValue()** para obter o LUID (Locally Unique Identifier) do privilégio que se pretende ajustar
3. Invocar **AdjustTokenPrivileges()** para modificar os privilégios pretendidos no token
4. Efectuar as operações pretendidas (que exigem os novos privilégios)
5. Invocar **AdjustTokenPrivileges()** novamente para repôr os privilégios anteriores
6. Fechar o handle para o token

Segurança em WindowsNT/Win32

Exemplo 1

Modificar a hora do sistema

➔ Slides seguintes.

Exemplo 2

Mudar o dono de um ficheiro

➔ Versão mais completa em documento à parte

Exemplo 3

Criar uma directoria em nome de um utilizador

➔ Documento à parte

Exemplo 4

Aceder a um named pipe remoto

➔ Partilhado com o assunto dos *named pipes* - Documento à parte

Segurança em WindowsNT/Win32

Exemplo: modificar o relógio do sistema

```
HANDLE      hToken;        /* token do processo        */
TOKEN_PRIVILEGES tp;       /* privilegios do token     */
TOKEN_PRIVILEGES oldtp;    /* priv.anteriores do token */
DWORD    dwSize = sizeof (TOKEN_PRIVILEGES);
LUID     luid;

/* obter privilégio SE_SYSTEMTIME_NAME para o processo */

if (!OpenProcessToken (GetCurrentProcess(), TOKEN_ADJUST_PRIVILEGES
     | TOKEN_QUERY, &hToken)) {
   printf ("OpenProcessToken() falhou com o código %d\n",
           GetLastError());
   return 1;
}
```

Segurança em WindowsNT/Win32

```
if (!LookupPrivilegeValue (NULL, SE_SYSTEMTIME_NAME, &luid)) {
   printf ("LookupPrivilege() falhou com o código %d\n",
           GetLastError());
   CloseHandle (hToken);
   return 1;
}
// Modificar o Token – acrescentar o privilégio pretendido
ZeroMemory (&tp, sizeof (tp));
tp.PrivilegeCount = 1;           // quantos privilégios a modificar
tp.Privileges[0].Luid = luid;       // qual priviégio
tp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;   // ligar

// ajusta (modifica) o privilégio pretendido
if (!AdjustTokenPrivileges (hToken, FALSE, &tp,
     sizeof(TOKEN_PRIVILEGES), &oldtp, &dwSize)) {
   printf ("AdjustTokenPrivileges() falhou com o código %d\n",
           GetLastError());
   CloseHandle (hToken);
   return 1;
}
```


Segurança em WindowsNT/Win32

```
/* Modifica o relógio  - usa função específica para essa acção */

if (!SetSystemTime (&stCurrentTime)) {
   printf ("SetSystemTime() falhou com código %d\n",
           GetLastError());
   CloseHandle (hToken);
   return 1;
}

/* repõe privileg. anteriores (retira SE_SYSTEMTIME_NAME)   */

AdjustTokenPrivileges (hToken, FALSE, &oldtp, dwSize, NULL,
                       NULL);
if (GetLastError() != ERROR_SUCCESS) {
   printf ("AdjustTokenPrivileges() falhou com código %d\n",
           GetLastError());
   CloseHandle (hToken);
   return 1;
}

CloseHandle (hToken);
```

Segurança em WindowsNT/Win32

Exemplo 2: modificar o dono de um ficheiro

Nota: este exemplo está melhor detalhado em documento à parte (no moodle)

```
BOOL SetFileOwner(LPSTR UserName, LPSTR FileName) {

     _SID_NAME_USE SIDType;
     char          Domain[2048];
     DWORD         dwDomainLength = 250;
     char          UserSID[1024];
     DWORD         dwSIDBufSize=1024;

     // obtem SID do novo user
     if (!LookupAccountName(
          NULL,                  // nome do sistema
          UserName,              // nome do utilizador
          UserSID,               // buffer para SID
          &dwSIDBufSize,         // tam. Inicial / final do SID
          Domain,                // dominio do sistema (saida)
          &dwDomainLength,       // tam. do dominio (saida)
          &SIDType)              // tipo do SID (saida)
        ) return FALSE;
```


Segurança em WindowsNT/Win32

```
     // adiciona o privilégio SE_RESTORE_NAME

     HANDLE hToken;
     LUID luid;
     TOKEN_PRIVILEGES tkp;

     // obtem handle para o access token

     if (!OpenProcessToken(
          GetCurrentProcess(),
          TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY,
          &hToken))
        return FALSE;

     // obtém valor local do privilégio
     if (!LookupPrivilegeValue(
                NULL,             // nome do sistema
                SE_RESTORE_NAME,  // privilégio
                &luid))           // valor local do privilégio
        return FALSE;
```

Segurança em WindowsNT/Win32

```
// adiciona o privilégio SE_RESTORE_NAME (cont)

tkp.PrivilegeCount = 1;
tkp.Privileges[0].Luid = luid;
tkp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;

if(!AdjustTokenPrivileges(
     hToken,        // token alvo
     FALSE,         // desligar todos=false
     &tkp,          // novo estado
     sizeof(TOKEN_PRIVILEGES),
     NULL,          // buffer estado anterior
     NULL))         // tam. Buffer estado anterior
   return FALSE;
}
```


Segurança em WindowsNT/Win32

```
// variáveis para SD actual
char                ucSDbuf[SD_SIZE];
PSECURITY_DESCRIPTOR pFileSD =
                          (PSECURITY_DESCRIPTOR)ucSDbuf;
DWORD               dwSDLengthNeeded;

// variáveis para novo SD
   char    NewSD[SECURITY_DESCRIPTOR_MIN_LENGTH];
PSECURITY_DESCRIPTOR psdNewSD =
                       (PSECURITY_DESCRIPTOR)NewSD;

// 1: obtem SD do ficheiro (mas não o vai usar neste exemplo)
   if(!GetFileSecurity(
     FileName,
     OWNER_SECURITY_INFORMATION,    // info pretendida
     pFileSD,                       // ptr para o SD
     SD_SIZE,
     &dwSDLengthNeeded))
   return(FALSE);
```

Segurança em WindowsNT/Win32

```
// 2: Inicializa novo SD
if(!InitializeSecurityDescriptor(psdNewSD,
    SECURITY_DESCRIPTOR_REVISION))
   return FALSE ;

// 3: Coloca novo SID no novo SD
if(!SetSecurityDescriptorOwner(psdNewSD,UserSID,0))
   return FALSE;

// 4: coloca novo SD no ficheiro
if (!SetFileSecurity(FileName,
     OWNER_SECURITY_INFORMATION, psdNewSD))
   return FALSE;

return TRUE;
```

DEIS/ISEC Sistemas Operativos 2 – 2021/22 João Durães, José Luís Nunes

27