МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Безопасность информационных и автоматизированных систем»

API – ФУНКЦИИ ДЛЯ РАБОТЫ С ТОКЕНОМ ДОСТУПА, ДЕСКРИПТОРОМ БЕЗОПАСНОСТИ, ACE

Методические указания
к выполнению лабораторной работы
по дисциплине «Безопасность операционных систем»
для студентов специальностей 090105, 090303, 090900

Курган 2012

Кафедра: «Безопасность информационных и автоматизированных систем»

Дисциплина: «Безопасность операционных систем» (специальность 090105, 090303, 090900).

Составил: ст. преподаватель А.Г. Рабушко

Утверждены на заседании кафедры « 28 » марта 2011 г.

Рекомендованы методическим советом университета « 26 » мая 2011 г.

ПРИБОРЫ, ОБОРУДОВАНИЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1 Операционная система (Microsoft Windows Xp).
2 Используемая среда программирования (Visual Studio C++).

ЦЕЛЬ РАБОТЫ

Программирование списков управления доступом.

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

Информация о правах доступа к любому объекту ядра находится в структуре **SECURITY_DESCRIPTOR.** Ее определение из файла winnt.h :

```
typedef struct _SECURITY_DESCRIPTOR {
  BYTE  Revision;
  BYTE  Sbz1;
  SECURITY_DESCRIPTOR_CONTROL Control;
  PSID Owner;
  PSID Group;
  PACL Sacl;
  PACL Dacl;
  } SECURITY_DESCRIPTOR, *PISECURITY_DESCRIPTOR;
```

Более конкретно, права доступа определяются в элементе **Dacl** указанной структуры. Определение структуры списка управления доступом:

```
typedef struct _ACL {
   BYTE  AclRevision;
   BYTE  Sbz1;
   WORD   AclSize;
   WORD   AceCount;
   WORD   Sbz2;
} ACL;
typedef ACL *PACL;
```

Хотя ACL может быть пустой (что предотвращает любой доступ), обычно он состоит из элементов, называемых ACE, которые явно либо разрешают, либо запрещают доступ:

```
typedef struct _ACE_HEADER {
   BYTE  AceType;
   BYTE  AceFlags;
   WORD   AceSize;
} ACE_HEADER;
```

```
typedef ACE_HEADER *PACE_HEADER;

typedef struct _ACCESS_ALLOWED_ACE {
   ACE_HEADER Header;
   ACCESS_MASK Mask;
   DWORD SidStart;
} ACCESS_ALLOWED_ACE;

typedef ACCESS_ALLOWED_ACE *PACCESS_ALLOWED_ACE;


typedef struct _ACCESS_DENIED_ACE {
   ACE_HEADER Header;
   ACCESS_MASK Mask;
   DWORD SidStart;
} ACCESS_DENIED_ACE;
typedef ACCESS_DENIED_ACE *PACCESS_DENIED_ACE;
```

Элемент **SidStart** определяет субъекта,а элемент **Mask** - предоставляемый доступ.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Для выполнения лабораторной работы необходимо:
Получить по имени пользователя/группы соответствующий sid, что реализуется вызовом функции LookupAccountName:

```
BOOL WINAPI LookupAccountName(
  LPCTSTR lpSystemName,
  LPCTSTR lpAccountName,
  PSID Sid,
  LPDWORD cbSid,
  LPTSTR ReferencedDomainName,
  LPDWORD cchReferencedDomainName,
  PSID_NAME_USE peUse
);
```

Произвести инициализацию необходимых структур:

```
BOOL WINAPI InitializeSecurityDescriptor(
  __out         PSECURITY_DESCRIPTOR pSecurityDescriptor,
  __in          DWORD dwRevision
);

BOOL WINAPI InitializeAcl(
```

```
  __out        PACL pAcl,
  __in         DWORD nAclLength,
  __in         DWORD dwAclRevision
);
```

Затем добавить в ACL соответствующие ACE с помощью функций AddAccessDeniedAce или AddAccessAllowedAce, которые имеют абсолютно одинаковый прототип:

```
BOOL WINAPI AddAccessDeniedAce(
  __in_out     PACL pAcl,
  __in         DWORD dwAceRevision,
  __in         DWORD AccessMask,
  __in         PSID pSid
);
```

После формирования ACL необходимо посредством вызова функции SetNamedSecurityInfo изменить DACL файла или директории, имя которых передается из командной строки:

```
DWORD WINAPI SetNamedSecurityInfo(
  __in         LPTSTR pObjectName,
  __in         SE_OBJECT_TYPE ObjectType,
  __in         SECURITY_INFORMATION SecurityInfo,
  __in_opt     PSID psidOwner,
  __in_opt     PSID psidGroup,
  __in_opt     PACL pDacl,
  __in_opt     PACL pSacl
);
```

Где в качестве pObjectName используется **argv[1]**, а параметр SecurityInfo устанавливается с помощью константы:
DACL_SECURITY_INFORMATION.

Для манипуляций со специфическими правами можно воспользоваться следующими опрелелениями:

```
#define FILE_READ_DATA           ( 0x0001 )    // file & pipe
#define FILE_LIST_DIRECTORY      ( 0x0001 )    // directory
#define FILE_WRITE_DATA          ( 0x0002 )    // file & pipe
#define FILE_ADD_FILE            ( 0x0002 )    // directory
#define FILE_APPEND_DATA         ( 0x0004 )    // file
#define FILE_ADD_SUBDIRECTORY    ( 0x0004 )    // directory
#define FILE_CREATE_PIPE_INSTANCE ( 0x0004 )    // named pipe
#define FILE_READ_EA             ( 0x0008 )    // file & directory
```

```
#define FILE_WRITE_EA           ( 0x0010 )    // file & directory
#define FILE_EXECUTE            ( 0x0020 )    // file
#define FILE_TRAVERSE           ( 0x0020 )    // directory
#define FILE_DELETE_CHILD       ( 0x0040 )    // directory
#define FILE_READ_ATTRIBUTES    ( 0x0080 )    // all
#define FILE_WRITE_ATTRIBUTES   ( 0x0100 )    // all
```

А также, при необходимости использования стандартных прав –

```
#define DELETE                  (0x00010000L)
#define READ_CONTROL            (0x00020000L)
#define WRITE_DAC               (0x00040000L)
#define WRITE_OWNER             (0x00080000L)
#define SYNCHRONIZE             (0x00100000L)
```

СПИСОК ЗАДАНИЙ:

1 Запрет группе Администраторы права исполнения файла.
2 Запрет группе Пользователи права чтения файла.
3 Разрешение зарегистрированному пользователю права читать владельца файла.
4 Разрешение зарегистрированному пользователю права записи владельца файла.
5 Разрешение группе Администраторы получения списка файлов в каталоге.
6 Разрешение группе Администраторы создания нового файла в каталоге.
7 Запрет Системе права чтения файла.
8 Разрешение Системе создавать новый файл в каталоге.
9 Разрешение всем полного доступа к файлу.
10 Запрет всем права чтения аттрибутов.
11 Разрешение субъектам сетевого входа права чтения аттрибутов.
12 Запрет субъектам интерактивного входа права исполения файла.

СПИСОК ЛИТЕРАТУРЫ

1 Руссинович М., Соломон Д. Внутреннее устройство Microsoft Windows. Мастер-класс.- СПб.: Изд-во «Питер», 2005.
2 Дейтел Х.М., Дейтел П.Дж, Чофнес Д.Р. Операционные системы. Ч. 2: Распределенные системы, сети, безопасность. – М.: Бином, 2006.
3 Дейтел Х.М., Дейтел П.Дж, Чофнес Д.Р. Операционные системы. Ч. 1: Основы и принципы. – М.: Бином, 2006.
4 Гордеев А.В. Операционные системы: учебник для вузов. – СПб.: Питер, 2004. – 416 с.
5 Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб.: Питер, 2001. – 544 с.
6 Кастер, Х. Основы Windows NT и NTFS. Русская редакция. – М., 1996.
7 Проскурин, В.Г. Защита в операционных системах. – М.: Радио и связь, 2000.

Рабушко Артур Германович

API – ФУНКЦИИ ДЛЯ РАБОТЫ С ТОКЕНОМ ДОСТУПА, ДЕСКРИПТОРОМ БЕЗОПАСНОСТИ, ACE

Методические указания
к выполнению лабораторной работы
по дисциплине «Безопасность операционных систем»
для студентов специальностей 090105, 090303, 090900

Редактор Е.А. Устюгова

Подписано к печати	Формат 60×84 1/16	Бумага тип. №1
Печать трафаретная	Усл. печ.л. 0,5	Уч.-изд.л.0,5
Заказ	Тираж э/в	Цена свободная

РИЦ Курганского государственного университета.
640669, г. Курган, ул. Гоголя, 25.
Курганский государственный университет.