HW 3 - Coding Question (#7)
James Foti
Red ID: 820124143
CS 574 - Computer Security
Professor Song
3/27/2020

7. Coding question (20 Points) Write a program that can display a file's list of security permissions and assign a list of permissions to a file.

Description –
Your program should be able to read the already existing file's permission and then change the permissions of the same file. One way of reading the file security information is by using system calls such as GetSecurityInfo or GetNamedSecurityInfo. The system call reads the list of permissions for an object. You must describe in short each system call that your program uses.

Reference - http://timgolden.me.uk/python/win32_how_do_i/add-security-to-a-file.html (http://timgolden.me.uk/python/win32_how_do_i/add-security-to-a-file.html)

NOTE: The reference above povides the solution to this problem. I am simply just documenting and describing the function calls that are being used!

In [1]:
```
import os,sys
import win32api
import win32security
import ntsecuritycon as con
```

In [2]:
```
FILENAME = "temp.txt"
os.remove(FILENAME)
```

In [3]:
```
def show_cacls (filename):
    print()
    print()
    for line in os.popen ("cacls %s" % filename).read ().splitlines ():
        print(line)
```

win32security.LookupAccountName() accepts the name of a system (string) and an account as input (int). It retrieves the SID for the account and the name of the domain where the account was found.

For more info: http://timgolden.me.uk/pywin32-docs/win32security__LookupAccountName_meth.html (http://timgolden.me.uk/pywin32-docs/win32security__LookupAccountName_meth.html)

In [4]:
```
#
# Find the SIDs for Everyone, the Admin group and the current user
#
everyone, domain, type = win32security.LookupAccountName("", "Everyone")
admins, domain, type = win32security.LookupAccountName("", "Administrators")
user, domain, type = win32security.LookupAccountName("", win32api.GetUserName ())
```

In [5]:
```
#
# Touch the file and use CACLS to show its default permissions
# (which will probably be: Admins->Full; Owner->Full; Everyone->Read)
#
open(FILENAME, "w").close ()
show_cacls(FILENAME)
```

```
C:\Users\james\OneDrive\Documents\GitHub\CS574_Computer_Security\Assignments\Assignment3
_File_Security_Permissions\temp.txt NT AUTHORITY\SYSTEM:F

BUILTIN\Administrators:F

DESKTOP-7T67BTL\james:F
```

win32security.GetFileSecurity() accepts the name of a file (string) and a flag that specifies the info requested (int). This returns the info about the file and is constrained by the caller's access rights.

For more info: http://timgolden.me.uk/pywin32-docs/win32security__GetFileSecurity_meth.html (http://timgolden.me.uk/pywin32-docs/win32security__GetFileSecurity_meth.html)

In [6]:
```
#
# Find the DACL part of the Security Descriptor for the file
#
sd = win32security.GetFileSecurity (FILENAME, win32security.DACL_SECURITY_INFORMATION)
```

AddAccessAllowedAce() accepts a revision (int), access (int), SID (int) as inputs and adds an access-allowed ACE to an DACL object.

More info: http://timgolden.me.uk/pywin32-docs/PyACL__AddAccessAllowedAce_meth.html (http://timgolden.me.uk/pywin32-docs/PyACL__AddAccessAllowedAce_meth.html)

In [7]:
```
#
# Create a blank DACL and add the three ACEs we want
# We will completely replace the original DACL with
# this. Obviously you might want to alter the original
# instead.
#
dacl = win32security.ACL()
dacl.AddAccessAllowedAce(win32security.ACL_REVISION, con.FILE_GENERIC_READ, everyone)
dacl.AddAccessAllowedAce(win32security.ACL_REVISION, con.FILE_GENERIC_READ | con.FILE_GE
NERIC_WRITE, user)
dacl.AddAccessAllowedAce(win32security.ACL_REVISION, con.FILE_ALL_ACCESS, admins)
```

SetSecurityDescriptorDacl() sets info in a discretionary access control list (DACL).

For more info: https://docs.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-setsecuritydescriptordacl (https://docs.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-setsecuritydescriptordacl)

win32security.SetFileSecurity() accepts a filename (string), info (int), security (PySECURITY DESCRIPTOR) and sets the infor about the security of a file or directory and is constrained by the caller's access rights.

For more info: http://timgolden.me.uk/pywin32-docs/win32security__SetFileSecurity_meth.html (http://timgolden.me.uk/pywin32-docs/win32security__SetFileSecurity_meth.html)

In [8]:
```
#
# Put our new DACL into the Security Descriptor,
# update the file with the updated SD, and use
# CACLS to show what's what.
#
sd.SetSecurityDescriptorDacl(1, dacl, 0)
win32security.SetFileSecurity(FILENAME, win32security.DACL_SECURITY_INFORMATION, sd)
show_cacls(FILENAME)
```

```
C:\Users\james\OneDrive\Documents\GitHub\CS574_Computer_Security\Assignments\Assignment3
_File_Security_Permissions\temp.txt Everyone:(special access:)


READ_CONTROL

SYNCHRONIZE

FILE_GENERIC_READ

FILE_READ_DATA

FILE_READ_EA

FILE_READ_ATTRIBUTES


DESKTOP-7T67BTL\james:(special access:)


READ_CONTROL

SYNCHRONIZE

FILE_GENERIC_READ

FILE_GENERIC_WRITE

FILE_READ_DATA

FILE_WRITE_DATA

FILE_APPEND_DATA

FILE_READ_EA

FILE_WRITE_EA

FILE_READ_ATTRIBUTES

FILE_WRITE_ATTRIBUTES


BUILTIN\Administrators:F
```