

Security

Estimated time for completion: 1 hour

Overview:

Goals:

* Enter goal(s) for this lab.

Lab Notes:

Enter any special setup-issues etc for the lab..

Use Principals to protect code layers

It is important to add security to every layer in your application. .NET can manage some aspects of this for you using Principal and Role objects. In Security1-Before, the application calls the methods GuestAccess and AdminAccess successfully. We want to prevent users from accessing AdminAccess.

Criteria:

* Enter criteria for the part here.

Notes:

* Enter any relevant notes for the part here.

Steps:

1. Create a GenericIdentity object with your name.
2. Create a GenericPrincipal object and add the role “Guest”.
3. Assign the GenericPrincipal object to the current thread.
4. Annotate the methods in RestrictedLayer with the PrincipalPermissionAttribute. Require the role “Guest” for GuestAccess. Require the role “Admin” for AdminAccess.
5. Run the program. It should fail with a SecurityException.

Compare files using hashes

An efficient way to compare two files is to compute their hash values and compare for equality. This is used for by many applications to verify that data and file systems have not been tampered with.

Steps:

1. Compute the hash value for both files (args[0] and args[1]) using the MD5 class.
2. Compare the two hash values, byte for byte. If the hash is different, print a message. Otherwise, print a message that the files are equal.
3. Use the PrintHash method to print a hexadecimal representation of the byte array.
4. Run the program on a text file: run file1 file1. It should say they are equal. Now copy that file and change just one character in the file. Run again. It should print very different MD5 values.

Solutions

Enter path(s) to solution file(s) here.