

RE: Do all three permission classes (Identity Permission, Code Access Permission and Role Based Permission) fall under CAS?

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2004-02/0387.html>

From: Shawn Farkas (shawdfa_at_online.microsoft.com)

Date: 02/26/04

Date: Thu, 26 Feb 2004 02:18:49 GMT

1. That is correct — the inputs to CAS for each assembly are that assembly's evidence and the current security policy. The output is that assembly's grant set.
2. Correct, only permissions which derive from System.Security.CodeAccessPermission are CAS permissions. Role based permissions do not derive from this class. Although there is a distinction between FileIOPermission and StrongNameIdentityPermission, I'm not sure that I would classify them as a code access permission and an identity permission, since StrongNameIdentityPermission is also a code access security permission (ie it derives from CodeAccessPermission).
3. Every permission can restrict access through declarative or imparative demands. One instance of using a It is correct that no request can ever exceed the permissions granted by the operating system. Just because I have a FullTrust assembly doesn't mean that I can read every file on the disk.
4. Identity permissions are a part of CAS. The reason I said that these could be interchangeable terms goes back to your first question. Say I have a policy that grants FileIOPermission to the c:\windows directory to any code from Microsoft.com. When I download an assembly from Microsoft.com, the site evidence is evaluated against that policy to produce a permission set containing (at least), a FileIOPermission and a SiteIdentityPermission. The document you read is differentiating between granting permissions and demanding them. I would consider both of these to be part of CAS.
5. Which three operations do O'Reilly define to be required of CAS? Note, that just because an object A supports the operations of object B does not mean that A is a B. For instance, an apple supports being eaten. A potato also supports being eaten. A potato is not an apple however. Since role based security objects do not derive from CodeAccessPermission, they are not CAS permissions.

.dotnet.security: RE: Do all three permission classes (Identity Permission, Code Access Permission and Role Based Perm

Hopefully that helped to clarify some more of your confusion ... again, if you have more questions, I'd be happy to keep answering :-)

-Shawn

<http://blogs.msdn.com/shawnfa>

--

This posting is provided "AS IS" with no warranties, and confers no rights.

Note: For the benefit of the community-at-large, all responses to this message are best directed to the person who originated.

>Thread-Topic: Do all three permission classes (Identity Permission, Code Access Permission and R
>thread-index: AcP7/QcaujX7TbQxQ4O2zH0FMsGQ8g==
>X-Tomcat-NG: microsoft.public.dotnet.security
>From: =?Utf-8?B?Tm92aWNl?=<6tclATqlinkDOTqueensuDOTca>
>References: <9384B26B-312C-4321-A85F-34FDB069BC86@microsoft.com> <4KbYYk9#DHA.616@cpmsftngxa06.
>Subject: RE: Do all three permission classes (Identity Permission, Code Access Permission and R
>Date: Wed, 25 Feb 2004 16:11:06 -0800
>Lines: 46
>Message-ID: <44A3D24D-2E2E-4B15-919E-299851646FBB@microsoft.com>
>MIME-Version: 1.0
>Content-Type: text/plain;
> charset="Utf-8"
>Content-Transfer-Encoding: 7bit
>X-Newsreader: Microsoft CDO for Windows 2000
>Content-Class: urn:content-classes:message
>Importance: normal
>Priority: normal
>X-MimeOLE: Produced By Microsoft MimeOLE V6.00.3790.0
>Newsgroups: microsoft.public.dotnet.security
>Path: cpmsftngxa06.phx.gbl
>Xref: cpmsftngxa06.phx.gbl microsoft.public.dotnet.security:5147
>NNTP-Posting-Host: tk2msftcmt1.phx.gbl 10.40.1.180
>X-Tomcat-NG: microsoft.public.dotnet.security
>

>Believe it or not, I've actually read a number of technical documents from Microsoft on .NET security references.

For clarification:

1.

Would you consider the Code Access Security policy system to be the system that takes evidence as input and produces a set of permissions based on those? The common trend I've seen in Microsoft function that takes two input variables and gives the output of permissions.

2.

a)

Instead of saying "Three Permission Classes" I should say there are "Three Types of Permission Classes" (Code Access Permissions, Identity Permissions, Role Based Permissions).

b)

In addition, only Code Access Permissions and Identity Permissions should be thought of as belonging to Code Access Security. However, Role Based Permissions do NOT belong to Code Access Security.

3.

a)

Through the use of Declarative and Imperative Security requests/statements, ONLY two types of permission classes (code access permission classes and identity permission classes) can restrict access to protected operations like creating files and directories. However, these requests can never allow the user to exceed the permissions that the OS has provided to them by the OS (hence the whole idea of .NET being built around the OS - I have a nice diagram that displays this). Is that correct?

b)

Through the use of Evidence and the configurable security policy Identity Permission classes can restrict access to protected operations like creating, deleting and modifying files and directories. However, these requests can never allow the user to exceed the permissions that the user has provided to them by the OS (hence the whole idea of .NET being built around the OS - I have a nice diagram that displays this). Is that correct?

RE: Do all three permission classes (Identity Permission, Code Access Permission and Role Based Permissions)

.dotnet.security: RE: Do all three permission classes (Identity Permission, Code Access Permission and Role Based Perm

4.

a)

You have implied that CAS and Evidence Based Security are different terminology for the same concept. Does evidence based security only refer to CAS provided through the use of Identity Permissions?

b)

Also in one of the Microsoft documents I read***, it differentiated between Evidence-based Security

Evidence-Based Security and Code Access Security

Two separate technologies work together to protect managed code:

Evidence-based security determines what permissions to grant to code.

Code access security checks that all code on the stack has the necessary permissions to do something.

Permissions bind these technologies together: a permission is the right to perform a specific process.

is a file permission; "to connect to www.msn.com" is a network permission.

[there is more information - please see the document I provide the link to at the bottom of this

which is different from what I would deduce it is from its wording (I stated my deduction in 4.a)

5.

You have said that Role-Based Permission classes do not belong to Code Access Security - however, in Evidence-Based Security, page 102 it says "CAS supports the following three permission request operations...."

in conjunction with Role-Based Permissions? Therefore, wouldn't Role Based Permissions be part of Code Access Security?

I have more questions - but if you are able to answer these questions above for me then I will be satisfied.

across too aggressively, I'm just trying to better understand how all of these concepts interrelate.

their relationship to one another I find myself more confused than before.

Thanks very much,

Novice

*** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/seccodeguide.a>

>

RE: Do all three permission classes (Identity Permission, Code Access Permission and Role Based Permissions)