

# Re: Windows versus Application Security

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.windowsforms/2006-08/msg00171.h>

---

- *From:* WhiskyRomeo <[WhiskyRomeo@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:WhiskyRomeo@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 20 Jul 2006 12:51:02 -0700
- 

Yep that is what I did.

I created this simple Class:

```
Public Class User
  Shared m_UserName As String

  Public Property UserName() As String
  Get
  Return m_UserName
  End Get
  Set(ByVal Value As String)
  m_UserName = Value
  End Set

  End Property

  Public Function GetUserName(ByVal sUser As String, ByVal sPassword As
  String) As String
  Return Me.UserName
  End Function
  End Class
```

"Patrice" wrote:

You don't have this kind of problem in a Windows application as the application is always alive. You can easily keep whatever values you want in a static/shared members of a class.

<http://msdn.microsoft.com/msdnmag/issues/05/04/Security/> goes far beyond and is perhaps a bit overkill but reading it may raise few ideas (basically the idea is to use the ASP.NET 2.0 security architecture from your Windows application).

—  
Patrice

Re: Windows versus Application Security

"WhiskyRomeo" <WhiskyRomeo@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> a écrit dans le message de news: A268F72B-28AE-49D3-A9DD-B0968ED40148@xxxxxxxxxxxxxxxxxxxx

You are correct. The question is how to maintain this globally in a Windows application. I know how to do this in a Web application using a Session variable.

What is the equivalent in a Windows Application?

wR

"Patrice" wrote:

It would be then some kind of applicative authorization.

So you would just have a login entry from that check the user likely from an account list stored in the DB. This identity is kept global in the windows Application and is passed to the server as needed (where it can be recorded or matched with the account list).

As a side note I'm afraid it could suffer from the same problem (if they already pass their credentials along to peers for Windows, why not for a custom application as they 'll have anyway also to manage accounts inside the application, and it seems they don't want to bother doing this for Windows).

Good luck.

--  
Patrice

"WhiskyRomeo"  
<WhiskyRomeo@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> a écrit dans le message de news: 1502BE0C-06A5-4FD1-B4B2-2C2C20FE3D2D@xxxxxxxxxxxxxxxxxxxx

Actually the only purpose of this is to capture who does what in the

## Re: Windows versus Application Security

application. There are already various status tables the capture who does what but is based on the windows login right which does not identify the actual individual. For example:

When an order is placed, an entry is made into the OrderStatus table which contains the Order\_ID, Status (in this case -- Placed), date, and user identification.

We most likely will continue to use their built in windows identity to control access to the database. So when Willy Wonka logs into his work station he logs in as PCUser. But when he opens the application, he must login as Willy Wonka and that identity must be passed around for the purpose of recording entries in these status tables.

WR

"Patrice" wrote:

Another option could be to use roaming profiles <http://support.microsoft.com/kb/243420/en-us> allowing the profile to be always available and allowing deletion server side... Looks like more a question for an admin group...

I'm not sure for the other part what you are trying to do ? Do you have a SQL Server 2000

Re: Windows versus Application Security

application that doesn't  
relate to this or do you  
mean  
you  
could have to create one to  
workaround this profile  
issue ?

--

Patrice

"WhiskyRomeo"

<WhiskyRomeo@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

a écrit dans le

message de news:

3CC25F36-6451-46AD-8DBF-90B598C06C7C@xxxxxxxxxxxxxxxxxxxx

I have a  
client that  
wants me to  
set up  
security for  
an windows  
.NET  
application  
using SQL  
Server 2000  
as the  
DBMS.  
Currently  
we use the  
built  
in  
Windows  
security to  
define login  
and access  
to the  
database.

The  
problem is  
that there is  
a lot of  
turnover  
and users  
work at  
multiple  
XP  
workstations.  
So everyone

## Re: Windows versus Application Security

logins under  
a common  
user name  
and  
password.

Creating,  
managing  
and  
removing  
windows  
domain  
accounts are  
not the  
problem.

The  
problem is  
that every  
time a new  
user logs on  
an XP  
workstation,  
that  
user's folder  
structure is  
created on  
that  
machine.

So, event  
though  
individual  
windows  
accounts are  
manageable,  
having to go  
through  
each  
workstation  
and clear  
out the  
users  
folders  
when they  
leave is not  
manageable  
— not to  
mention the  
waste of  
disk space.

Is there a

## Re: Windows versus Application Security

way to  
prevent XP  
from  
creating the  
individual  
user's  
folder  
structure for  
each user?

If I must  
create an  
application  
login  
system,  
what is the  
best way  
to  
do  
this? That is  
how do I  
pass the  
identity of  
the user  
from one  
form  
to  
another?

WR