

Re: Access 2003 adp/proxy security – A substitute for SYSTEM_USER()

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.connect/2004-11/0176.html>

From: Kimberley Yochum (*kyochum_at_wthq.com*)

Date: 11/23/04

Date: Tue, 23 Nov 2004 16:01:31 -0500

Thank you Aaron, but all the user functions (System_user, current_user, session_user, user, user_name, suser_sname, etc.) return the Proxy user name once a SQL user login has been performed.

Still hoping for some understanding about what Access is doing to my connection string values or suggestions on what we can do in SQL...

Thank you in advance.

"aaron kempf" <aarkem@safeco.com> wrote in message
news:OF9GDhZ0EHA.2228@TK2MSFTNGP15.phx.gbl...

> *i dont want to sound dumb, but you've tried SUSER_SNAME() and all the
other*

> *ones, right??*

>

> *also, if you're scanning against a large list of Text data, you might want*

> *to consider using HASH INDEXES (aka the checksum function) in order to
speed*

> *this.*

>

>

>

>

> *"Kimberley Yochum" <kyochum@bigzoo.net> wrote in message*

> *news:%23BOvPMX0EHA.632@TK2MSFTNGP10.phx.gbl...*

>> *We are moving an Access2000 adp application to Access 2003. Access2000*

>> *version uses approle for security but we found Approle does not work the*

>> *same in Access2003 so we are switching to a proxy security method.*

>>

>> *Problem*

>>

>> *We need a way for SQL to know the nt username that initiated the proxy*

>> *connection. We need SQL to be able to retrieve that username very, very*

>> *quickly (basically without a table lookup) for each user's spids,*

> *regardless*

> > *of how many connections Access decides to make for the adp.*
> >
> > *Details:*
> >
> > *a.. All the insert and update triggers depended on SYSTEM_USER to stamp*
> > *the user who wrote the data on the row*
> > *b.. Many stored procedures and views use a UDF that also depends on*
> > *SYSTEM_USER*
> > *c.. Since we're reconnecting each user as a proxy user, we need a way*
> > *for*
> > *SQL to determine the nt user of the application.*
> > *d.. Note: Modifying the client to pass the user in is simply not*
> > *practical*
> > *due to the huge amount of code change.*
> > *Our solution*
> >
> > *We are using SET CONTEXT_INFO to stuff the user's name into the*
> > *context_info*
> > *of sysprocesses when the app first logs in. We have a UDF – fnSystemUser*
> > *that queries sysprocesses to return what SYSTEM_USER used to.*
> > *Unfortunately,*
> > *for the triggers or any UDFs (which previously only hit SYSTEM_USER),*
> > *that*
> > *meant a nose dive in performance since the UDF is executing on every row*
> > *instead of evaluating once as if it were deterministic for the life of*
> > *that*
> > *query. Although SYSTEM_USER is nondeterministic, it appears it was only*
> > *evaluated once for the queries instead of the performance we see now on*
> > *our*
> > *UDF returning the system user name from context_info, that indicates*
> > *executing on every row.*
> >
> > *Since Access adps open multiple connections dynamically our fnSystemUser*
> > *is*
> > *coded to find the Context_info of the spid that was first set through*
> > *client*
> > *code. See the code below:*
> >
> >
> > *CREATE FUNCTION dbo.fnSystemUser()*
> > *RETURNS nvarchar(50)*
> > *AS*
> > *BEGIN*
> >
> > *DECLARE @ContextInfo varbinary(128)*
> > *DECLARE @DomainUserName nchar(128)*
> >
> > *-- First attempt to get the username stuffed into*
> > *context_info of this connection*
> >

microsoft.public.sqlserver.connect: Re: Access 2003 adp/proxy security – A substitute for SYSTEM_USER()

```
>> SELECT @ContextInfo = context_info
>> FROM master..sysprocesses
>> WHERE spid = @@SPID
>>
>> -- Convert it to nvarchar
>> SET @DomainUserName = CAST(CAST(REPLACE(@ContextInfo,0x0000,"")
AS
>> varbinary(128)) AS nvarchar(50))
>>
>> -- If the context info is blank then we're on one of the
> connections
>> -- that Access dynamically created but the client code can't
access
>> to stuff
>> -- something into the context_info. Interrogate all other spids
> for
>> this
>> -- user and this client process (using net_address to guarantee
> we've
>> got the
>> -- right user), to pull username from the context info of another
>> connection/spid
>>
>> IF LEN(@DomainUserName) < 1
>> BEGIN
>> DECLARE @NetAddress nchar(12)
>>
>> SELECT TOP 1 @NetAddress = net_address
>> FROM master..sysprocesses p
>> WHERE SPID = @@SPID
>>
>> SELECT TOP 1 @ContextInfo = context_info
>> FROM master..sysprocesses p
>> INNER JOIN master..sysdatabases d
>> on p.dbid = d.dbid
>> WHERE hostprocess = HOST_ID()
>> AND d.Name = DB_NAME()
>> AND NOT CAST(context_info as nchar(50))= SPACE(50)
>> AND (LEN(@NetAddress) < 1 OR net_address = @NetAddress)
>> ORDER BY Context_info
>>
>> SET @DomainUserName =
>> CAST(CAST(REPLACE(@ContextInfo,0x0000,"") AS varbinary(128)) AS
>> nvarchar(50))
>>
>> END
>> /* Fail safe, if we couldn't find a non-empty Context_Info to
>> discover the
>> logged in user then could be the user is attached with Query
>> Analyzer or
>> some other method and we'll just use the user as they are
```

Re: Access 2003 adp/proxy security – A substitute for SYSTEM_USER()

logged

> > *into SQL */*

> > *IF LEN(@DomainUserName) < 1*

> > *SET @DomainUserName = SYSTEM_USER*

> >

> > *RETURN*

> >

>

SUBSTRING(@DomainUserName,CHARINDEX(N'\',@DomainUserName)+1,LEN(@DomainUserN

ame)-(CHARINDEX(N'\',@DomainUserName)))

> > *END*

> >

> > *GO*

> >

> > *Solving the performance problem*

> >

> > *Here are some workaround we've tried:*

> >

> > *a.. Querying sysprocesses seems slow so we instead created a table*

> > *SyUserLogin and use HostID() and HostName() to find the right row. We've*

> > *seen SQL change HostName (or Workstation ID) to "Pool08" and seen 2*

users

> > *have the same HostName. So there is no guarantee that when a user logs*

in

> > *that HostID() and HostName() will uniquely identify each user. but the*

> > *odds*

> > *are in our favor.*

> >

> > *Basically here's what runs (stripped of the uniqueness checks) when*

the

> > *user first logs in to SQL:*

> >

> > *INSERT INTO [dbo].[SyUserLogin]*

> > *([UserName], [HostName], [HostID], [NetAddress])*

> > *SELECT @SystemUserName, -- username that we pass in*

> > *Host_Name(),*

> > *Host_id(),*

> > *@NetAddress -- from the sysprocesses row for this spid*

> >

> > *Here's the new fnSystemUser:*

> >

> > *CREATE FUNCTION dbo.fnSystemUser()*

> > *RETURNS nvarchar(50)*

> > *AS*

> > *BEGIN*

> > *RETURN*

> > *(SELECT UpdateUserName*

> > *FROM SyUserLogin*

> > *WHERE HostName = Host_NAME()*

> > *AND HostID = HOST_ID()*

> > *)*

```
> > END
> >
> >
> > a.. We tried to change the connection string to use either
"Application
> > Name" or "Workstation ID" as a cubby hole to stuff the nt user name. We
> > can
> > get this to work in a VB Script:
> > dim objConnection
> > dim strAccessConnect
> >
> > strAccessConnect = "Provider=SQLOLEDB.1" & _
> > ";Net=dbnmpntw;Data Source=ASQLServer" & _
> > ";Initial Catalog=MyDatabase" & _
> > ";Persist Security Info=False;Application
> > Name=MyTestApp;Workstation ID=NTUserName;"
> >
> > Set objConnection = createobject("ADODB.Connection")
> > objConnection.Open strAccessConnect, "OurProxy", "ProxyPassword"
> > MsgBox objConnection.ConnectionString
> > objconnection.close
> > set objconnection = nothing
> >
> >
> > But when we try this in Access2003, Access seems to stomp on those value
> > when it connects. We've played with the Connection dialog settings, but
> > can't seem to convince Access to leave our connection string values
alone.
> >
> > If we could get this to work then it would be ideal. Querying HostName()
> > or
> > even Appname() although obscure would be as fast as SYSTEM_USER()
> >
> > a.. Could we somehow create a function that would wrap context_info
and
> > match the speed of SYSTEM_USER?
> > Thank you in advance for reading through this and commenting!
> >
> > Kimberley Yochum kyochum@wthq.com
> >
> >
> >
> >
> >
> >
```