Issues with SQL Server Login Failures having dependency on Windows Password Policy Check!

Author : admin

Today users reported unique accessibility issues where all SQL logins are found locked out in all SQL servers (2005, 2008) where Windows enforce password policy is enabled, on further investigation it was found for each SQL login there are more than 20 failed attempts in SQL server error log. It took lot of effort in finding a root cause as on first attept it looked like a SQl server virus injection, but later found that some Windows level script is scheduled by security team which is checking whether SQL server has any default password configured or not :S anywys we are unale to find who configured it (should be some nerd who created more problem then solving anything), but it now execute every week on a specific time, so to fix this from SQL end, we disabled enforce password policy from all SQL servers in order for applciations to run smoothly.

So to fix it, found following script on Sudarshan blog

Script to disable Enforce password policy from all SQL logins:

```
DECLARE @name sysname
DECLARE @tmpname nvarchar (1024)
set @tmpname = ”
DECLARE login_curs CURSOR FOR
select name from sys.sql_logins where is_policy_checked = 1;

OPEN login_curs
fetch next from login_curs into @name
WHILE @@fetch_status = 0
BEGIN
print ‘Altering property CHECK_POLICY for Login: ‘ + @name
set @tmpname = N‘ALTER LOGIN ‘+@name+‘ WITH CHECK_POLICY = OFF’
print @tmpname
exec sp_executesql @tmpname
FETCH NEXT FROM login_curs INTO @name
END

CLOSE login_curs
DEALLOCATE login_curs
GO
```

On execution it disabled the enforce policy option and the life is smoother afterwards

To add onto it, from Sudarshan blog found few good facts, please find exceprts below, you can anyways read full blog post at following link

> **Symptoms**
>
> Typical symptoms you would notice are these:-
>
> a) Your SQL Server logins start failing
> b) SA account is unable to login
> c) Any SQL Agent jobs for which a SQL login is the owner start failing.
> d) Users/Application which use SQL authenticated logins are experiencing login failures.

For e.g., you might notice these error messages in the job history,

If you look up the SQL Error logs, you should see this,

```
2009-09-29 11:38:35.65 Logon Error: 18456, Severity: 14, State: 10.
2009-09-29 11:38:35.65 Logon Login failed for user ‘SqlLogin’. [CLIENT: 172.30.199.199]
```

Let me tell you something more about “State: 10”. This state is reached while evaluating the password policy checks. So any SQL Login which has is_policy_checked=1, are eligible for state 10 failures.

You might ask me as to how does this affect your SQL Logins??
It does! If the SQL Server service account gets locked out on the domain controller, all SQL Server authenticated logins which use domain password policy enforcement feature (CHECK_POLICY) will be unable to login to the SQL Server until the service account is “unlocked”.

And more importantly check_policy is by default ON for any SQL login you create, unless you explicitly turn it off.

So how does this check_policy work?
SQL Server after all is an application and it has to rely on the exposed Windows API’s to do work for it when it comes to any external authentication. This is implemented by call to WinAPI **NetValidatePasswordPolicy** to implement password complexity, history, lockout etc. Now, in SQL the thread which calls this API runs under the context of the SQLSvc account. So, if the SvcAccount is locked out, this operation fails & hence the login to SQL fails. Make sense ?

If you look up the documentation for the *NetValidatePasswordPolicy* API, it does 3 types of validation

NetValidateAuthentication (for checking password expiration and account lockout policy)
NetValidatePasswordChange (password validation when change password is done)
NetValidatePasswordReset (password validation during when password reset is done. Can also reset the lockout state)

Next up, check if all the accounts which owned jobs or experienced login failures have the check_policy turned **ON**. You can verify all the logins which have the check_policy turned on running the query,

```
select * from sys.sql_logins where is_policy_checked = 1
```

More importantly, since SQL 2005 SP2, microsoft has added new ring buffer entries (sys.dm_os_ring_buffers) for various security errors. For more info on this read RDorr’s blog at http://blogs.msdn.com/psssql/archive/2008/03/24/how-it-works-sql-server-2005-sp2-security-ring-buffer-ring-buffer-security-error.aspx

So, lets query the security ring buffer for more information,

```
select * from sys.dm_os_ring_buffers where ring_buffer_type = ‘RING_BUFFER_SECURITY_ERROR’
```

The Record column will contain the API Name as well as any ErrorCode that was returned by the API.

<u>Sample output with error</u>

```
RING_BUFFER_SECURITY_ERROR 3435668357

67
NetValidatePwdPolicy
CAPIPwdPolicyManager::ValidatePwdForLogin
0×8
```

As you can see a WinAPI call to NetValidatePwdPolicy is made to validate the password and there was a failure there. This is returned by GetLastError WinAPI. In the above logs the error returned was “0×8?
0×8 -> 8 (decimal) -> Not enough storage is available to process this command.

Interesting you should see this here. This means your system event logs should certainly have some kind of Kerberos/netlogon errors reported. Here is a sample,

```
11/11/2009 2:32:11 PM NETLOGON Error None 5719 N/A SFRDBC2 “This computer was not able to set up a secure
```

session with a domain controller in domain LITTLER-US due to the following: Not enough storage is available to process this command.

This may lead to authentication problems. Make sure that this computer is connected to the network. If the problem persists, please contact your domain administrator.

11/11/2009 2:32:11 PM Kerberos Error None 7 N/A SFRDBC2 The kerberos subsystem encountered a PAC verification failure. This indicates that the PAC from the client _besadmin in realm STARWARS.COM had a PAC which failed to verify or was modified. Contact your system administrator

These are some of the things you can do:-

1) Verify if the service account is locked out. If it is, then have your sysadmin unlock it. But you still need to unlock your SQL account or restart sql service for them to work again. Also thinking ahead you might want to audit and find out who/why locked out the service account.

Now this can happen for any of the following reasons :-

a) Invalid attempts (by someone or some application) using the SQLSvc account to login to Windows.
b) SvcAccount is disabled on the DC.
c) Password has expired for the SvcAccount etc.

2) If service account is NOT locked out and you are experiencing errors similar to ones above, the quick fix for this issue, is to disable CHECK_POLICY for the SQL Logins

In case you this is not an option and you want to avoid such issues in the future, we have introduced a new trace flag T4614. Trace flag 4614 when enabled allows SQL Server authenticated logins (eg. SA) that use Windows domain password policy enforcement (check_policy = ON) to log on to the instance even though the SQL Server service account is locked out or disabled on the Windows domain controller.

Note: this was introduced in build 2005.90.2194 and can be enabled as a dynamic trace flag using DBCC TRACEON (4614,-1). Read more about it at http://support.microsoft.com/default.aspx?scid=kb;EN-US;925744

3) Get your System/Active Directory administrator to look into the Kerberos warnings/failures to see what the issue is.

Hope you have found article useful and would have helped in solving some of your issues!