Using PowerShell to resolve Token Size issues caused by SID history

learn.microsoft.com/en-us/archive/blogs/ashleymcglone/using-powershell-to-resolve-token-size-issues-caused-by-sid-history

- Article
- 05/19/2011

Update: To see all articles in this series click here.

No matter how bad he wants to an elephant simply cannot drive a Volkswagen. He just won’t fit in that space. People can drive VWs, because that is who the VW was designed to accommodate. There is nothing worse than frustrated elephants. What does this have to do with PowerShell or Active Directory? I’m glad you asked.

The Problem: Token Size and SID History

When users become members of too many groups their access token grows so large that it no longer fits inside some of the default OS setting. Users can see issues logging in, applying group policy, and authenticating to web servers. There are several other good articles that explain this scenario, and they are linked at the end of this post.

Token size issues can usually be attributed to a combination of three scenarios:

1. SID history left over from AD migration projects
2. Poor group nesting strategy
3. Old groups that are never purged

Today we will address the SID history scenario with a PowerShell script that documents the extent of SID history in your environment and creates a SID mapping file for use with the ADMT to migrate resources to the new SIDs.

Tell me if this scenario has ever happened at your company. You are doing an AD migration with the ADMT (or a similar product). You migrated the users and groups. The project schedule was falling behind. You take a shortcut with the member servers. Instead of using the ADMT security translation process you just rejoin the servers to the new domain, leaving all of the old ACLs and SIDs untouched. You leave SID history in place and move on to the next big project and deadline.

Management is happy that you met the deadline, but little do they know that you created an entire zoo full of elephants whose VWs are starting to feel kind of tight.

This scenario is common, and I want to explain the risks here.

1. Down the road you are likely to encounter token size issues, because now all users and groups have at least two SIDs.

2. If you decide to clean up SID history in the future you have the potential of orphaning users from their data, because the old SIDs were still giving them access.
3. THE BIG ONE: When the project is decommissioned and the ADMT is blown away you lose all of the OldSID/NewSID migration data needed to clean the ACL SIDs on your migrated resource servers.

Many companies that I work with find themselves in this place. Users are calling the help desk with mysterious symptoms, and they are not sure what is causing the problem. Often it is token size, and SID history is a significant component. At this point you could purge the SID history, but then you could lose access to all of the migrated data with old SIDs in the ACLs. The best way forward is to go back and finish the AD migration by doing security translation on all of the migrated resources.

The Solution

The following steps will rid your environment of SID history:

1. Identify all servers in the environment that were around during the AD migration(s).
2. Install the latest version of the ADMT on a member server in your domain (NOT A DC, follow the guide linked below).
3. Run the PowerShell script from this article to create a SID mapping file.
4. Run the ADMT security translation wizard against each of the old servers, and use the SID mapping file when prompted to retrieve objects for security translation. View the ADMT log files to see where changes were made. (I started to write a PowerShell script to do the security translation, and I got it working for file share permissions by editing SDDL strings. That was pretty cool, but it would take too much time to rewrite all seven steps built into the ADMT wizard.) The ADMT wizard will clean SIDs in the following locations:
    1. Files And Folders
    2. Local Groups
    3. Printers
    4. Registry
    5. Shares
    6. User Profiles
    7. User Rights
5. In phases purge SID history from users and groups, starting with a small test population and then going by department until it is entirely removed from the environment. Use targeted LDAP query strings to feed the PowerShell script linked at the end of this article to do your purges. DO NOT PURGE ALL SID HISTORY AT ONCE. That could be a resume-generating event if you missed some resources in the re-migration. Do it in smaller batches to be safe. Rerun the script in this article to see where SID history remains.

Enter PowerShell. Since the ADMT database is long gone, we have to rebuild the SID mapping between old and new SIDs. Lucky for us this data is all present in AD, because you told the ADMT to keep it. We just need to put it into a file that the ADMT wants, called the SID mapping file. This file format is simply a CSV file where the first column contains the old SID and the second column contains either the new SID or the new user name in DOMAIN\USERNAME format. The script in today’s post will do this for you. This script is entirely safe for your environment, since it makes no changes and only reads data.

In addition to the mapping file the script will also generate a CSV report of all SID history in your domain. It includes the following columns: samAccountName, DisplayName, objectClass, OldSID, NewSID, DistinguishedName. Use this report before the clean up to assess the scope of the issue. Run it again after the clean up to make sure you got it all.

Another handy way to view SID history for a specific user is with NTDSUTIL. You can use the command Group Membership Evaluation to view all of a user’s groups, and it will tell you which ones come from SID history. This is another process to spot check SID history before and after the cleanup.

The Script

This script is rather simple really. Initially I thought you could dump all SID history with a PowerShell one-liner. And you may, but the trick is that the SID history attribute is a multi-value field. This syntax will not work:

```
Get-ADObject -LDAPFilter "(sidHistory=*)" -Property sidHistory | Export-CSV SIDHistory.csv
```

This will get you a query of the objects, but the SID history column on every row will say “Microsoft.ActiveDirectory.Management.ADPropertyValueCollection”.

Author's note: See the new-improved version of this script as a one-liner here.

To do this properly and make the report generation easy I used two loops. The outside loop iterates through all of the objects with SID history, and the inside loop iterates through each of the multi-value SID history attribute values. Then I capture the properties for the report into a custom object for easy manipulation. I create the custom object using a PowerShell technique called “splatting”. (Read more about that v2 feature over on the “Hey, Scripting Guy!” blog here.) This report treats the multi-value attributes as separate rows so that we get a one-to-one list of OldSID/NewSID. Finally I dump the SID mapping data into two separate CSV files for your convenience: a detailed report and an ADMT SID mapping file. For the ADMT mapping file we have to do a quick cleanup of the quotes, because the ADMT doesn’t like those in the input file.

```
<#-----------------------------------------------------------------------------
Get-SIDHistory.ps1
Ashley McGlone, Microsoft PFE
https://blogs.technet.com/b/ashleymcglone
April, 2011

This script queries Active Directory for SID history in order to build a SID
mapping file for use with the ADMT to do security translation, especially in
situations where the ADMT database has been lost. In addition to the mapping
file it also generates a full SID history report for viewing in Excel. This
script must be run from a machine that has the Active Directory module for
PowerShell installed (ie. Windows 7 or Windows Server 2008 R2 with RSAT).
You must also have either a Windows Server 2008 R2 domain controller, or an
older domain controller with the Active Directory Management Gateway Service
(AD Web Service) installed. For more information on ADMGS see:
https://aka.ms/ADPS2003
-----------------------------------------------------------------------------#>

Import-Module ActiveDirectory

#Create a blank array to hold our SID Map data
$arySIDMap = @()

#Query SID history, current SID, and related fields from AD
$ADQuery = Get-ADObject -LDAPFilter "(sIDHistory=*)" -Property objectClass, `
    samAccountName, DisplayName, objectSid, sIDHistory, distinguishedname

#Loop through each AD object returned
ForEach ($row in $ADQuery) {
    #SID history is a multi-valued attribute, so loop through each entry.
    ForEach ($SID in $row.sIDHistory) {
        #Arrange the data we want into a custom object
        $objTemp = New-Object PSObject -Property @{
             objectClass=$row.objectClass;
             OldSID=$SID;
             NewSID=$row.objectSID;
             samAccountName=$row.samAccountName;
             DisplayName=$row.displayName;
             DistinguishedName=$row.distinguishedName
            }
        #Use array addition to add the new object to our SID Map array
        $arySIDMap += $objTemp
    }
}

#Create a full SID History report file for reference in Excel
$arySIDMap | Export-CSV .\SID_History_Report.csv -NoTypeInformation

#Create a SID Mapping text file for use with ADMT
$arySIDMap | Select-Object OldSID, NewSID |
    Export-CSV .\SIDMapping1.csv -NoTypeInformation
#Peel out the quotes from the mapping file, because ADMT does not like those.
Get-Content .\SIDMapping1.csv | ForEach-Object {$_.Replace("`"","")} |
    Set-Content .\SIDMapping.csv
Remove-Item .\SIDMapping1.csv
```

```
"Output complete:"
"SID_History_Report.csv - full SID History report for reference in Excel"
"SIDMapping.csv - file for use with ADMT to do security translation"

# ><>
```

Conclusion

I hope this long-winded explanation resolves your SID history issues and informs your future AD migration projects. Who knew that PowerShell was a weight-loss product for elephants?! Now your tokens can fit inside the VW.

Links

Get-SIDHistory.p-s-1.txt

Comments

- **Anonymous**
  November 21, 2011
  If this article was helpful to you check out all of the newer posts on SID history: blogs.technet.com/.../sid+history

- **Anonymous**
  December 10, 2015
  This post is part four in the "PowerShell: SID Walker, Texas Ranger" series on documenting and remediating SID history in your AD forest. In today's post we will look at the final step of remediating SID history: removing the SID history

- **Anonymous**
  February 22, 2016
  After speaking about SID history and token size at PowerShell Saturday last month an attendee approached me with a common concern. I was so excited to code the answer that I did it in the airport on the way home.
  Joe User has been with the company

- **Anonymous**
  January 19, 2017
  These exercises can preserve skin from sagging to minimize the look of a double chin.

- **Anonymous**
  January 20, 2017
  To realize this, you will will need to consume about 600 fewer calories per day than your physique needs to preserve weight.

- **Anonymous**
  January 20, 2017
  Se você ingerir mais do que seu corpo precisa, ele não elimina excesso, este acumula e faz reserva singularmente na localidade abdominal, por isso é tão difícil emagrecer e também perder barriga.

- **Anonymous**
  January 30, 2017
  Para isso, é preciso conseguir acompanhar a lição e ter pique para intercalar a agilidade com ritmos fortes.