isi_acl

A Script to Modify Windows ACLs on Isilon Without an SMB Connection

Adam Fox, Isilon Corporate Advisory Engineer adam.fox@dell.com

Introduction

Isilon is a scalable storage system that is the basis for a data lake. This means that the system supports many different protocols to access a common data set. With respect to permissions, Isilon currently supports both POSIX bits as well as Windows ACLs. When a Windows ACL is present, that permission is enforced even if that ACL cannot be properly expressed in the displayed POSIX bits. This limits what clients that cannot handle Windows ACLs can view and modify with respect to file permissions. The goal of this script is to use a RESTful API provided by Isilon OneFS to view and manipulate Windows ACLs from clients that do not have an SMB connection to the cluster.

Support

Please note that this script is not officially supported by Dell EMC. This means that Isilon support cannot help with any questions or issues with the script. The author of the script will attempt to address any concerns but there is no SLA, timeframe, or commitment to do so. The script is, however, open source and anyone is welcome to update, fix, or modify the code as desired.

Setup

The script consists of two files, *isi_acl* and *papi.py*. The former is the code that is run by the user and the latter is a library to assist in making the API calls. Both must be present on the client in order to run the script properly. The script is written in Python 2.7.13. It has not been tested with Python 3 so it may be advisable to have a release close to 2.7.13 on the client. The rest of the Python libraries that are imported by the script are very common and should be standard on an installation of Python. If any are missing, then they will need to be installed in order for the script to run.

There are two things to consider in the isi_acl file before running the code. The first is to check the first line and ensure that it has the proper path to the Python interpreter on the client. The script comes with the path of */usr/local/bin/python*. Feel free to change that as needed. The second is a default cluster hostname if that is desired. Look for the line in the script that reads as follows:

```
cluster = "10.111.158.130"
```

This is the default IP that will be used if one is not specified on the command line with the –c flag. If it is not desired to always specify a cluster name/IP on the command line, this string can be changed to any name (including the SmartConnect zone name) or an IP address of a cluster. If no cluster is specified on the command line, this is what will be used by the script. It is not required to change this line, but it is there for convenience.

Syntax and Examples

The syntax for the script is as follows:

```
Usage: isi_acl [-ch] add|remove user|group allow|deny ace[,ace,ace,...] path
       isi_acl [-ch] add|remove everyone allow|deny ace[,ace,ace,...] path
       isi_acl [-ch] view path
-c | --cluster : Name or IP address of the cluster other than default
-h | --help : Prints this message
```

Note that the ace (or Access Control Entries) are specified in OneFS nomenclature as one would use if modifying the ACLs via the CLI on the cluster. See the man page for chmod(1) for details. The view function is written to resemble the view of running the ls –led command on the cluster. It is, therefore, useful to become familiar with these cluster commands when using this script.

To view the ACLs of a file, use the following syntax:

```
[foxa3@AFOX-51-c66-vm isi_acl]$ ./isi_acl view /ifs/data/Recovery/ny1.mov
User: root
Password:
-rwx------ +  1 AD2\foxa3  AD2\domain users  1567178 Feb 24 2006
/ifs/data/Recovery/ny1.mov
 OWNER: user:AD2\foxa3
 GROUP: group:AD2\domain users
 0: user:AD2\foxa3 allow file_gen_all
```

Note that a user and password must be provided. All API calls must be authenticated. Therefore a user that is defined on the cluster must be provided with each call. That user must be in one of the providers set up for the cluster, specifically in the System zone as the API only works in the System zone today. Also note that the user specified must have permissions to view or modify the file or the call will fail. This script is not a work-around for file and directory permissions. In the above example, the root user is given, but that is not required as long as the user that is supplied has the proper permissions to view or modify the file. Also note that the script requires the full path of the file or directory no the cluster so it should always start with */ifs*. As states earlier, the view subcommand is intended to resemble the output of the *ls –led* command on the cluster.

Now as an example of how to add a ACE to an ACL:

```
[foxa3@AFOX-51-c66-vm isi_acl]$ ./isi_acl –c isilon1 add group
'ad2\domain admins' allow file_gen_all /ifs/data/Recovery/ny1.mov
User: root
Password:
-rwxrwx--- +  1 AD2\foxa3  AD2\domain users  1567178 Feb 24 2006
/ifs/data/Recovery/ny1.mov
 OWNER: user:AD2\foxa3
 GROUP: group:AD2\domain users
 0: group:AD2\domain admins allow file_gen_all
 1: user:AD2\foxa3 allow file_gen_all
```

In this case, the –c flag was used to specify the cluster name ‘isilon1’. The command adds the ACE, then shows the result to the screen (i.e. it runs the view command under the covers).

There is an exception to using the 'user' or 'group' tag on the command and that is for the well-known group "Everyone". If you want to manipulate that ACE, the keyword 'group' can be left off as this is reserved in Windows.

Finally, this example will remove the ACE that was just created

```
[foxa3@AFOX-51-c66-vm isi_acl]$ ./isi_acl -c 10.111.158.130 remove group
'ad2\domain admins' allow file_gen_all /ifs/data/Recovery/ny1.mov
User: root
Password:
-rwx------ +  1 AD2\foxa3  AD2\domain users  1567178 Feb 24 2006
/ifs/data/Recovery/ny1.mov
 OWNER: user:AD2\foxa3
 GROUP: group:AD2\domain users
 0: user:AD2\foxa3 allow file_gen_all
```

A note for this syntax. It is common for some entries to have multiple permissions such as 'file_read,execute'. When deleting an entry, only one of the permissions need to be specified. An example follows:

```
[foxa3@AFOX-51-c66-vm isi_acl]$ ./isi_acl view /ifs/data/Recovery/ny1.mov
User: root
Password:
-rwxr-x--- +  1 AD2\foxa3  AD2\domain users  1567178 Feb 24 2006
/ifs/data/Recovery/ny1.mov
 OWNER: user:AD2\foxa3
 GROUP: group:AD2\domain users
 0: group:AD2\domain admins allow file_read,execute
 1: user:AD2\foxa3 allow file_gen_all


[foxa3@AFOX-51-c66-vm isi_acl]$ ./isi_acl remove group 'ad2\domain admins'
allow file_read /ifs/data/Recovery/ny1.mov
User: root
Password:
-rwx------ +  1 AD2\foxa3  AD2\domain users  1567178 Feb 24 2006
/ifs/data/Recovery/ny1.mov
 OWNER: user:AD2\foxa3
 GROUP: group:AD2\domain users
 0: user:AD2\foxa3 allow file_gen_all
```

Note the command only needed one of the permissions to match and delete the entry.