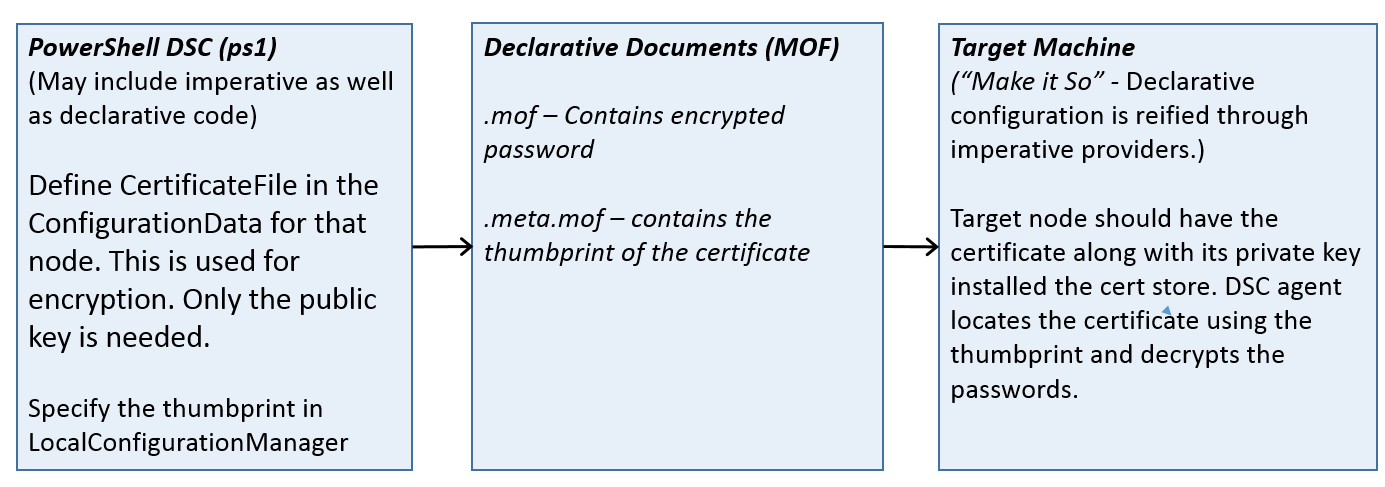
PowerShell DSC Password Encryption Hello world Example

Sivaprasad Padisetty (June 2014)

*PowerShell DSC (Desired State Configuration) is new in Windows Server 2012 R2/Windows 8.1 and is part of WMF 4.0 (Servers Windows Management Framework). WMF 4.0 can be installed in other versions of Windows. It can be downloaded from* [*here*](http://www.microsoft.com/en-us/download/details.aspx?id=40855)*. When DSC script (ps1) is executed an intermediate document (MOF) is produced which contains all the declarative state including passwords and is persisted on disk. Obviously you want the passwords to be encrypted. This blog gives a simple example for encrypted passwords.*

# Basic flow

DSC is an extension to PowerShell language. This means it is authored as a ps1 file, which can contain both imperative and declarative constructs. When such a ps1 script is executed it produces a declarative document (MOF). If the ps1 file references password, it uses the defined certificate file to encrypt. (i.e.) The generated MOF file only has the encrypted password. The certificate along with its private key should be installed in the target node by other means (outside of DSC). DSC agent in the target node locates the right certificate using the thumbprint. Because the private key is present, the password can be decrypted.



# Certificate Setup

Given it is a test, we will be using a self-signed test certificate with subject ‘CN=DSC Test Certificate’. Below script checks if such a certificate is already installed in the store. If not it searchers for makecert.exe to create a self-signed certificate. If it cannot find makecert.exe in the path, it tries to locate the pfx file. The pfx file is expected to have poor man’s password “password”. In the github location below the script along with a test pfx file is published for convenience. By end of executing below script $cert will have the certificate object, $certfile will point to the certificate file with public key.

In this script to make it simple, generation of MOF and application of MOF happens in the same machine.

#Check if the test certificate is already installed

# Subject for this certificate should be 'CN=DSC Test Certificate'

$cert = dir Cert:\LocalMachine\My | where { $\_.Subject -eq 'CN=DSC Test Certificate' }

if (-not $cert)

{

# Certificate not found, so will try to create one

if (gcm makecert.exe 2>$null)

{

#generates certificate and installs in the My store

makecert -r -pe -n "CN=DSC Test Certificate" -sky exchange -ss my -sr localMachine

}

else

{

#If a pfx file is present in $PSScriptRoot or in current directory

#import it into My Store

#following commands are used to create pfxfile

#It is precreated so script will run without makecert dependency

#makecert -sky exchange -n "CN=DSC Test Certificate" -pe -sv "DSC Test Certificate.pvk" "DSC Test Certificate.cer"

#pvk2pfx -pvk "DSC Test Certificate.pvk" -spc "DSC Test Certificate.cer" -pfx "DSC Test Certificate.pfx" -pi password

#Locate pfxfile

$pfxfile = dir "$psscriptroot\\*DSC Test Certificate.pfx" | select -First 1

if (-not $pfxfile)

{

$pfxfile = dir ".\\*DSC Test Certificate.pfx" | select -First 1

}

if ($pfxfile)

{

$password = ConvertTo-SecureString -String "password" -Force –AsPlainText

Import-PfxCertificate -FilePath $pfxfile -Exportable -Password $password `

-CertStoreLocation Cert:\Localmachine\My

}

else

{

throw 'Did not find DSC Test Certificate, please install makecert.exe in the path'

}

}

$cert = dir Cert:\LocalMachine\My | where { $\_.Subject -eq 'CN=DSC Test Certificate' }

}

$certfile = 'c:\temp\testfile\DSC Test Certificate.cer'

Export-Certificate -Cert $cert -FilePath $certfile

# DSC Script (ps1)

Since everything is done in the local machine, for convenience the nodename is defined as localhost. This means it will run in any machine and need not match specific computer name. Executing this script will produce localhost.mof and localhost.meta.mof.

#Config definition

configuration main

{

node "localhost"

{

$password = "Secret.1" | ConvertTo-SecureString -asPlainText -Force

$username = "user1"

$cred = New-Object System.Management.Automation.PSCredential($username,$password)

User u1

{

UserName = "$username";

Password = $cred;

}

LocalConfigurationManager

{

CertificateID = $cert.Thumbprint

}

}

}

$config=

@{

AllNodes = @(

@{

NodeName = "localhost"

CertificateFile=$certfile

};

);

}

# Invoke main to generate MOF

main -OutputPath C:\temp\config -ConfigurationData $config

# localhost.mof

/\*

@TargetNode='localhost'

@GeneratedBy=Administrator

@GenerationDate=06/07/2014 23:29:49

@GenerationHost=TEST

\*/

instance of MSFT\_Credential as $MSFT\_Credential1ref

{

Password = "9hLGCVMNA5bYBaOT2T18AEyot8ut3aa6WEZUtMqmIBjZHs4mMdHLMtXNziMFbiyes4L05YldRWbY62iVLCIOAbVj39DI0KmEnvKXogH/GNvFHrptwGLq1Ox2sddk5xiymr/sjKriFShZuWdfKV9HUEoinBGJPI2zjUgh4fI2VkUDJVdzG0m/GxwQ1WsWlKtdqU8wrHQFkEWDzAzYY+LWRHA7vaVn8zLMw2U9guHsk7b9SO3jSNYE3ehbv9ZAPYqgaPfao8R3guxwnQbdWd5ROYi/X/9Eu2+uEXvc7vkTC2lkOUc3Mg3VO3GKa9xBiDSfBb71j6gD6/haRWnsjjCKaw==";

UserName = "user1";

};

instance of MSFT\_UserResource as $MSFT\_UserResource1ref

{

ResourceID = "[User]u1";

UserName = "user1";

Password = $MSFT\_Credential1ref;

SourceInfo = "C:\\temp\\8 - Encryption.ps1::64::9::User";

ModuleName = "PSDesiredStateConfiguration";

ModuleVersion = "1.0";

};

instance of OMI\_ConfigurationDocument

{

Version="1.0.0";

Author="Administrator";

GenerationDate="06/07/2014 23:29:49";

GenerationHost="TEST";

ContentType="PasswordEncrypted";

};

# localhost.meta.mof

/\*

@TargetNode='localhost'

@GeneratedBy=Administrator

@GenerationDate=06/07/2014 23:29:49

@GenerationHost=TEST

\*/

instance of MSFT\_DSCMetaConfiguration as $MSFT\_DSCMetaConfiguration1ref

{

CertificateID = "428E1C2320968C20D12B64AA3A2EF743522E068F";

};

instance of OMI\_ConfigurationDocument

{

Version="1.0.0";

Author="Administrator";

GenerationDate="06/07/2014 23:29:49";

GenerationHost="TEST";

};

# Applying MOF

# Push the MOF file generated to the target node.

Set-DscLocalConfigurationManager -ComputerName localhost -Path C:\temp\config

# Push the MOF file generated to the target node.

Start-DscConfiguration c:\temp\config -ComputerName localhost -Wait -verbose -Force

# Code

The code can be found under “DSC” folder at <https://github.com/padisetty/Samples>.

Explore & Enjoy!

/Siva