

```

1  #Get-SANEncryptionKey.ps1
2  #Written by: Jeff Brusoe
3  #Last Updated: May 8, 2020
4  #
5  #The purpose of this file is to backup the SAN encryption keys.
6
7  <#
8  .DESCRIPTION
9      This file backs up the SAN encryption keys and emails them out every morning.
10
11      It requires the installation of Unisphere CLI found in this directory
12      (UnisphereCLI-ForBackups.exe) .
13
14  .PARAMETER BackupDirectory
15      Specifies which directory to store the lbb files in.
16
17  .PARAMETER PasswordFile
18      This is the encrypted file that is used to store the password in order to send out
19      the
20      backup key emails.
21
22  .NOTES
23      Filename: Get-SANEncryptionKey.ps1
24      Author: Jeff Brusoe
25      Last Updated By: Jeff Brusoe
26      Last Updated: August 26, 2019
27  #>
28
29  [CmdletBinding()]
30  param (
31      #Common Parameters
32      [switch]$NoSessionTranscript,
33      [string]$LogFilePath = "$PSScriptRoot\Logs\",
34      [switch]$StopOnError, #true is used for testing purposes
35      [int]$DaysToKeepLogFiles = 7, #this value used to clean old log files
36
37      #File specific parameters
38      [string]$BackupDirectory = "$PSScriptRoot\EncryptionKeys\",
39      [string]$PasswordFile = "GSE3.txt", #used to send email
40      [switch]$Prompt,
41      [switch]$ChangeSecureStringFile,
42      [switch]$Testing #Just used as a testing flag for parts of code
43  )
44
45  #####
46  # Initialize Environment #
47  #####
48  #Build path to HSC PowerShell Modules
49  $PathToHSCPowerShellModules = $PSScriptRoot
50  $PathToHSCPowerShellModules =
51  $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
52  $PathToHSCPowerShellModules += "1HSC-PowerShell-Modules"
53  Write-Output $PathToHSCPowerShellModules
54
55  $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psm1"
56  Write-Output "Path to common code module: $CommonCodeModule"
57
58  Import-Module $CommonCodeModule -Force -ArgumentList
59  $NoSessionTranscript,$LogFilePath,$StopOnError,$DaysToKeepLogFiles
60
61  Write-Verbose "Getting Parameter Information"
62  Get-Parameter -ParameterList $PSBoundParameters
63
64  Set-Environment
65  Set-WindowTitle
66
67  #Remove old key files

```

```

64 Write-Verbose "Removing old encryption keys"
65 Remove-OldLogFile -LBB -Path $BackupDirectory -Days 14 -Verbose -Delete
66
67 #####
68 # End of Environment Configuration Block #
69 #####
70
71 #Configure email parameters
72 $Recipients =
73 "jbrusoe@hsc.wvu.edu","krodney@hsc.wvu.edu","rnicols@hsc.wvu.edu","microsoft@hsc.wvu.edu",
74 "mlogue@hsc.wvu.edu"
75
76 #The following block of code calls the Unisphere Command Line Interface (uemcli) and
77 are based on an email from Kim.
78 #Documentation for the uemcli utility can be found here:
79 https://www.emc.com/collateral/TechnicalDocument/docu69330.pdf.
80
81 #These are flags to indicate success/failure and are mainly used to generate the final
82 email.
83 $Success102 = $false
84 $Success133 = $false
85
86 Write-Output -"Beginning to backup keys for 10.3.4.102"
87 Write-Output "Backup Directory: $BackupDirectory`n"
88
89 #Decrypt password to connect to SAN
90 $PasswordFile = "$PathToHSCPowerShellModules\$PasswordFile"
91 Write-Output "Password File: $PasswordFile"
92 $Password = Get-PasswordFromSecureStringFile -Prompt $Prompt -ChangeSecureStringFile
93 $ChangeSecureStringFile -PasswordFile $PasswordFile
94 $EmailPassword = $Password.substring(0,$Password.Length - 2) + "!"
95
96 $Error.Clear()
97 Uemcli -d 10.3.4.102 -u Local/systems -p $Password -download -d $BackupDirectory
98 encryption -type backupKeys | tee "102.txt"
99
100 if ($Error.Count -gt 0)
101 {
102     Write-Warning "There was an error generating encryption key for 10.3.4.102"
103     $Error
104 }
105
106 if ((select-string -path "102.txt" -Pattern "operation completed successfully") -ne
107 $null)
108 {
109     Write-ColorOutput -ForegroundColor "Green" -Message "`nKeys backed up for
110 10.3.4.102.`n"
111     $Success102 = $true
112 }
113 else
114 {
115     Write-Error "`nError backing keys up for 10.3.4.102`n"
116
117     Send-MailMessage -To $Recipients -from "microsoft@hsc.wvu.edu" -Subject "Error
118 backing up encryption keys for 10.3.4.102" -SmtpServer "hssmtp.hsc.wvu.edu"
119 -Priority High -Verbose
120 }
121
122 Write-ColorOutput -ForegroundColor "Green" -Message "`nBeginning to backup keys for
123 10.3.4.133"
124 Write-ColorOutput -ForegroundColor "Green" -Message $("Backup Directory:
125 $BackupDirectory`n")
126
127 $Error.Clear()
128 Uemcli -d 10.3.4.133 -u Local/systems -p $Password -download -d $BackupDirectory
129 encryption -type backupKeys | tee "133.txt"

```

```

117
118 if ($Error.Count -gt 0)
119 {
120     Write-Warning "There was an error generating encryption key for 10.3.4.133"
121 }
122
123 if ((select-string -path "133.txt" -Pattern "operation completed successfully") -ne
124 $null)
125 {
126     Write-ColorOutput -ForegroundColor "Green" -Message "`nKeys backed up for
127     10.3.4.133.`n"
128     $Success133 = $true
129 }
130 else
131 {
132     Write-Error "`nError backing keys up for 10.3.4.133`n"
133
134     Send-MailMessage -To $Recipients -from "microsoft@hsc.wvu.edu" -Subject "Error
135     backing up encryption keys for 10.3.4.133" -SmtpServer "hssmtp.hsc.wvu.edu"
136     -Priority High -Verbose
137 }
138
139 if ($Success102 -AND $Success133)
140 {
141     Get-ChildItem -path $BackupDirectory
142     $Attachments = Get-ChildItem -path $BackupDirectory | Sort-Object -Property
143     LastWriteTime -Descending | select FullName -first 2 | foreach {$_.FullName}
144
145     Write-Output "Attachments:"
146     Write-Output $Attachments
147
148     $Subject = (Get-Date -format yyyy-MM-dd) + " Successfully backed up SAN encryption
149     keys"
150     $MsgBody = "Encryption keys are stored on sysscript3 in $BackupDirectory and are
151     also attached to this email."
152
153     try
154     {
155         Send-MailMessage -Body $MsgBody -To $Recipients -From "microsoft@hsc.wvu.edu"
156         -Subject $Subject -Attachments $Attachments -SmtpServer "hssmtp.hsc.wvu.edu"
157         -Verbose -ErrorAction Stop
158         Write-Output "Successfully sent email"
159     }
160     catch
161     {
162         Write-Warning "There was an error attempting to send the email"
163         Write-Output $Error | FL
164     }
165 }
166
167 Write-Verbose "Cleaning up files."
168 Remove-Item "102.txt"
169 Remove-Item "133.txt"
170
171 Write-Verbose "Stopping session transcript"
172 if (!$NoSessionTranscript)
173 {
174     Stop-Transcript
175 }
176
177 #Exit

```