

```
1 <#
2 .SYNOPSIS
3     The purpose of this file is to set extensionAttribute14 to flag users who
4     should
5     take the hospital's HIPAA training.
6
7 .DESCRIPTION
8     The users who take the hospital's HIPAA training are sent to the microsoft
9     account
10    every morning around 8am. This email has a subject of "IdentityIQ Report".
11    For the time being, this file is manually saved to the
12    \HSC-PowerShell-Repository\2CommonFiles\HospitalHIPAATraining directory.
13    A file is being worked on that will automatically save files like this sent by
14    email.
15
16    Once the file is saved, this file goes through and sets extensionAttribute14
17    for those users.
18
19 .PARAMETER HospitalFilePath
20     Specifies the path to search for the latest hospital HIPAA file
21
22 .PARAMETER Testing
23     Used for additional logging features if development work is being done on this
24    file.
25
26 .NOTES
27     Set-HospitalHIPAATraining.ps1
28     Last Modified by: Jeff Brusoe
29     Last Modified: May 24, 2021
30 #>
31
32 [CmdletBinding()]
33 param (
34     [ValidateNotNullOrEmpty()]
35     [string]$HospitalFilePath = $((Get-HSCGitHubRepoPath) +
36         "2CommonFiles\HospitalHIPAATraining"),
37     [switch]$Testing
38 )
39
40 #Configure environment
41 try {
42     Set-HSCEnvironment -ErrorAction Stop
43     Write-Output "Hospital File Path: $HospitalFilePath"
44 }
45 catch {
46     Write-Warning "Unable to configure environment"
47     Invoke-HSCExitCommand -ErrorCount $Error.Count
48 }
49
50 #Geneate list of users from hospital file
51 if (Test-Path $HospitalFilePath)
52 {
53     $HospitalFile = Get-HSCLastFile -DirectoryPath $HospitalFilePath
54     Write-Output "Hospital File: $HospitalFile"
55
56     try {
57         $WVUMUsers = Import-Csv $HospitalFile -ErrorAction Stop
58     }
59 }
```

```

55     catch {
56         Write-Warning "Unable to open hospital file. Program is exiting."
57         Invoke-HSCEExitCommand -ErrorCount $Error.Count
58     }
59 }
60 else {
61     Write-Warning "Unable to access file from hospital. Program is exiting."
62     Invoke-HSCEExitCommand -ErrorCount $Error.Count
63 }
64
65 $FoundCount = 0
66 $NotFoundCount = 0
67
68 #Now loop through list of users
69 foreach ($WVUMUser in $WVUMUsers)
70 {
71     $Username = $WVUMUser."User ID".Trim()
72     Write-Output $("Username: " + $Username)
73
74     $LDAPFilter = "(samaccountname=$Username)"
75     Write-Output "LDAPFilter: $LDAPFilter"
76
77     try
78     {
79         $GetADUserParams = @{
80             LDAPFilter = $LDAPFilter
81             Properties = "extensionAttribute14"
82             ErrorAction = "Stop"
83         }
84
85         $ADUser = Get-ADUser @GetADUserParams
86
87         if ($null -ne $ADUser) {
88             Write-Output "User Found"
89             Write-Output $("Sam Account Name: " + $ADUser.SamAccountName)
90             Write-Output $("extensionAttribute14: " + $ADUser.extensionAttribute14)
91         }
92     }
93     catch {
94         Write-Warning "Unable to find AD user"
95         $ADUser = $null
96     }
97
98     if ($null -eq $ADUser) {
99         Write-Output "User not found"
100         $NotFoundCount++
101     }
102     elseif ($ADUser.extensionAttribute14 -eq "HospitalHIPAA") {
103         Write-Output "extensionAttribute14 is already set."
104         $FoundCount++
105     }
106     else {
107         try {
108             Write-Output "User found - Need to set extensionAttribute14"
109             $ADUser |
110             Set-ADUser -Clear extensionAttribute14 -ErrorAction Stop
111
112             Start-Sleep -s 1
113
114             $ADUser |

```

```
115     Set-ADUser -Add @{extensionAttribute14="HospitalHIPAA"} -ErrorAction Stop
116 }
117 catch {
118     Write-Warning "Error setting extension ext14"
119 }
120
121 $FoundCount++
122 }
123
124 Write-Output "*****"
125
126 if ($Error.Count -gt 0 -AND $Testing) {
127     Invoke-HSCEExitCommand -ErrorCount $Error.Count
128 }
129 }
130
131 Write-Output "Found: $FoundCount"
132 Write-Output "Not Found: $NotFoundCount"
133
134 Invoke-HSCEExitCommand -ErrorCount $Error.Count
```