

```
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". For
11    the time being,
12    this file is manually saved to the $PSScriptRoot/FilesFromHospital directory. A
13    file is being
14    worked on that will automatically save files like this sent by email.
15
16    Once the file is saved, this file goes through and sets extensionAttribute14
17    for those users.
18
19 .PARAMETER HospitalFilePath
20     Specifies the path to search for the latest hospital HIPAA file
21
22 .PARAMETER Testing
23     Used for additional logging features if development work is being done on this
24     file.
25
26 .DESCRIPTION
27     Users who should take the hospital HIPAA training are sent to the
28     microsoft@hsc.wvu.edu
29     account once per day around 8am. This file is stored in the FilesFromHospital
30     directory.
31
32 .NOTES
33     Set-HospitalHIPAATraining.ps1
34     Last Modified by: Jeff Brusoe
35     Last Modified: October 14, 2020
36 #>
37
38 [CmdletBinding()]
39 param (
40     [ValidateNotNullOrEmpty()]
41     [string]$HospitalFilePath = "$PSScriptRoot\FilesFromHospital\",
42     [switch]$Testing
43 )
44
45 #Configure environment
46 Set-HSCEnvironment
47 Write-Output "Hospital File Path: $HospitalFilePath"
48
49 #Geneate list of users from hospital file
50 if (Test-Path $HospitalFilePath)
51 {
52     $HospitalFile = Get-HSCLastFile -DirectoryPath $HospitalFilePath
53     Write-Output "Hospital File: $HospitalFile"
54
55     try {
56         $WVUMUsers = Import-Csv $HospitalFile -ErrorAction Stop
57     }
58     catch {
```

```

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

```

```
112     Invoke-HSCExitCommand -ErrorCount $Error.Count
113 }
114 }
115
116 Write-Output "Found: $FoundCount"
117 Write-Output "Not Found: $NotFoundCount"
118
119 Invoke-HSCExitCommand -ErrorCount $Error.Count
```