# Purpose

Enforce Multi Factor Authentication (MFA) compliance for all Licensed users in Office 365.

For each licensed user in Office 365, do the following steps;

1. Collect the following attribute of each user
   1. DisplayName
   2. UserPrincipalName
   3. Account Created Date
   4. Current StrongAuthentication (MFA) State
   5. The MFADefaultType, if user has registered their MFA preferences.
2. Make a list of all members of the MFA-Exclusion-Group, and take no further actions on those users
3. If the user has no MFADefaultType, instructions are emailed to the user and a log keeps count of number of notices sent.
4. If the user has an MFADefaultType, but is not in and Enabled or Enforced State, StrongAuthentication (MFA) is enabled and the user is emailed to expect a prompt to finish logging in to MFA.
   1. The date the user was enabled is logged.
5. If the day is Friday, an Admin report of all collected data for the previous 7 days is sent to a designated email address.

# Recommended Environment Specs

* Windows Server 2012 R2
* PowerShell 5.1, with PowerShell ISE
* .Net 4.5
* Internet Access
* Domain Service Account to run Script in Task Scheduler
  + Local Admin on Server
* O365 Account with 'User Account Administrator' to Set MFA settings in PowerShell (Microsoft says Global Admin is required, but this has been working for me)

# Setup

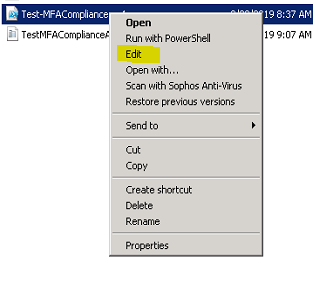
There are 4 files included;

1. Test-MFACompliance.ps1
2. PRG-Logo.png
3. FH-Logo.png
4. MFACompliance.docx (this document)

The logos are used for branding the Weekly Admin Report. Additional Agency Logos can be found here; <https://www.omnicomprgroup.com/agencies/>. You will need to resize the logo, for example, FH-Logo.png is Resized to 375 x 120.

Login to server as your service account and copy these files to C:\Scripts\MFACompliance, and use your own logo if desired. It is recommended to run this on a server dedicated to scripts and user management.

Right Click and Edit the script in ISE, and set the information in the top Section





While in $Test = $true mode, no changes are made to accounts, everything is informational only to ensure it is working.

Changes within quotes, must remain in quotes. Such as;

$TestTo = "whatever email address you want to use"

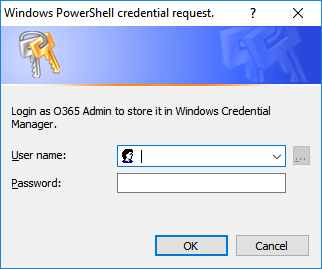
Set your SMTP relay, and Email Template to users (in HTML markdown); <https://www.markdownguide.org/basic-syntax/>

The script needs to be run in ISE as the Service Account for the Scheduled Task. This is needed to install the required Modules and store the Credentials of the Office 365 Admin Account in Windows Credential Manager.

Leave the $test variable as $true until you are confident everything is configured properly

$Test = $true

Run the Script and watch the console for information. Complete the Credential Prompt. If the Credentials are incorrect, a warning is output to the console and an email is sent to the $Admin.



In Test mode, the $TestTo variable is the email address that will receive ALL the actions of the script. If there are 300 notifications, this user will get all of them. Test mode also sends the Admin report to the $TestTo even if it is not Friday, and it opens it immediately in your default browser. These actions are to ensure that all parts of the script are going to function.

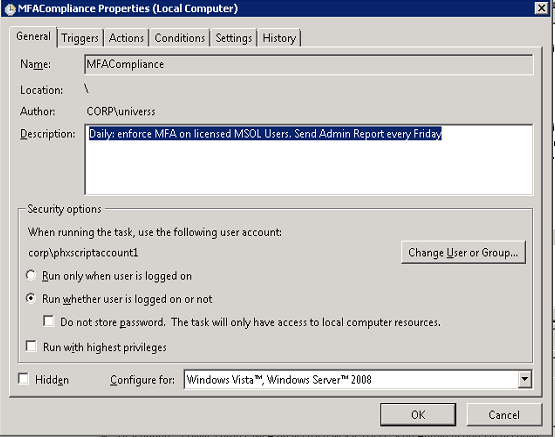
Some Actions will occur in the Script Directory;

1. TestMFAComplianceAdminLog.json is created to store information for the admin report and can be checked to make sure everything is working
2. MFAComplianceAdminLog.json is created and maintained for $Test = $False (Production Mode)
3. Log.Log is created to show a summary of actions each time the script is running. This is an appended log that always records all console output.
4. A \Reports folder is created to store the HTML Admin Reports. The test report is overridden, the weekly report is date indexed.

Once you are satisfied that the script is working as expected, change the $test variable to $false

$Test = $false

We use Task Scheduler to run the script daily at 10:00 AM Central.



Triggers Tab:

On a Schedule  
 Daily, at selected time  
 Recur every 1 day

Actions Tab:

Action: Start a Program  
 Program/script: powershell.exe  
 Arguments: -ExecutionPolicy Bypass C:\Scripts\MFACompliance\Test-MFACompliance.ps1  
 Start in: C:\Scripts\MFACompliance

It’s recommended to monitor the Log.log file for a few days to make sure no errors are occuring.