Microsoft 365 Business Powershell Runbook

Prepared by

Nick Ross

Microsoft Certified Expert Administrator

(msp4msps@tminus365.com)

Guide Description

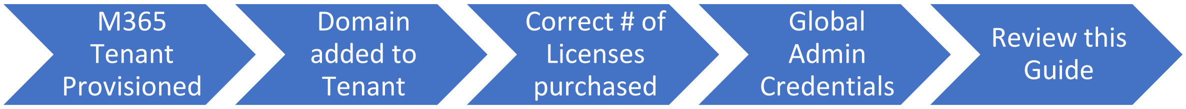
The purpose of this guide is to provide a powershell runbook for implementing M365 Business. This guide is assuming you have the ***M365 Business*** *License. After you run this powershell script you will have created/enabled:*

1. All users and groups with assigned licenses added
2. MFA for all users
3. Encryption Rules Set up to auto-apply for DLP
4. Azure Information Protection label for encryption of documents and emails
5. Advanced Threat Protection policies set up for safe links and safe attachments
6. Intune Configuration with the following:
   a. A device compliance policy for:
      i. iOS
      ii. Android
      iii. Windows
   b. A device configuration policy for Windows Devices to have BitLocker
   c. Terms and Conditions for when users enroll
   d. Office 365 Business pushed out as a required App to window 10 devices and uninstall existing versions of proplus
   e. Microsoft Authenticator pushed out as a required App for iOS and Android devices

Disclaimer

This guide is meant to provide best practices for policy creation and implementation of Microsoft 365 Business. It is meant to be used as a template, but the policies defined will not be the same in all use cases. You must access to policies and configuration you will need for your customers environment and make changes as needed. TMINUS is not liable for any policies you create that do not meet the customers standards. As a best practice, test all configurations with a pilot group before moving to broad deployment across an entire organization

Pre-Flight Checklist



a. M365 Tenant Provisioned
    i. Provision tenant through a CSP provider
b. Domain Added to Tenant
    i. Go to Setup>Domains and add the domain for the tenant you wish to use
    ii. Add TXT record to DNS settings to prove ownership
c. Correct # of Licenses purchased
    i. Ensure you have the correct number of licenses that match the user list you will be creating to upload
d. Ensure you have access to the Global admin credentials for the tenant
    i. You will be prompted for these credentials when you run the script
e. Review this Guide
    i. Ensure the settings outlined in this guide meet your requirements.
    ii. You can configure settings appropriately or remove certain features as necessary to customize your template
    iii. I will be walking through each configuration section in this guide so you know exactly what's being implemented

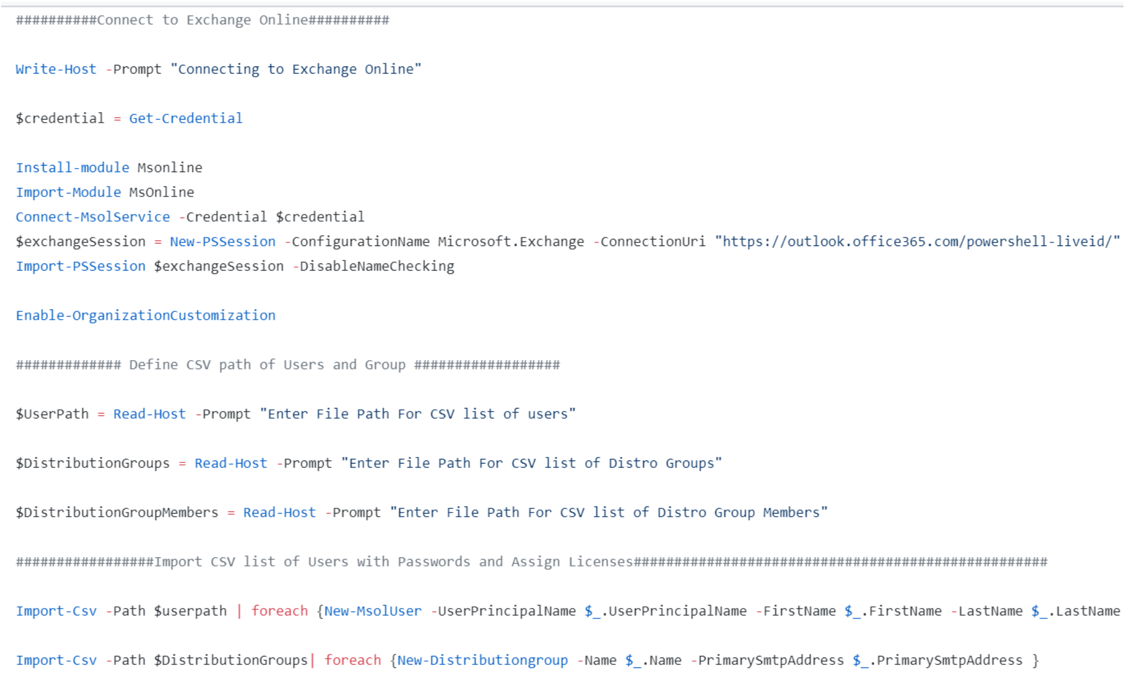
Where to view the guide

The script is published on github at the following link: https://github.com/msp4msps/M365-Buisness/blob/master/M365%20Business%20Runbook.ps1

I will be walking through each section of the script so you know what is all encompassed if you were to run this on any tenant

Create CSV of Users and Groups

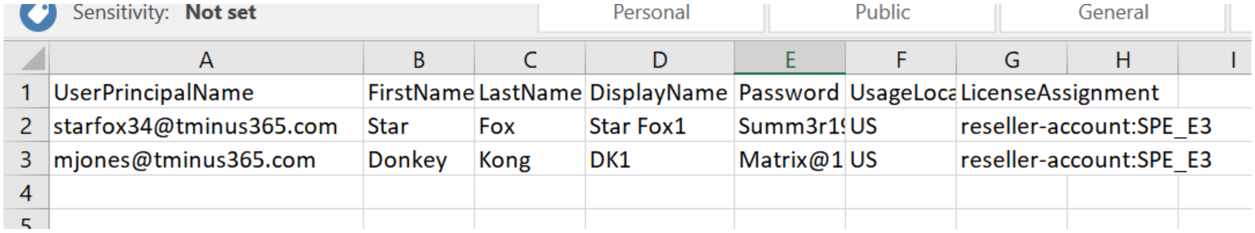
The first section of the script is connecting to exchange online and asking you the file paths of 3 CSV files which you will want to create. This part of the script defines your CSV paths and creates/licenses all of your users in the tenant as well as create 365 Distribution Groups



```
##########Connect to Exchange Online##########

Write-Host -Prompt "Connecting to Exchange Online"

$credential = Get-Credential

Install-module Msonline
Import-Module MsOnline
Connect-MsolService -Credential $credential
$exchangeSession = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri "https://outlook.office365.com/powershell-liveid/"
Import-PSSession $exchangeSession -DisableNameChecking

Enable-OrganizationCustomization

############# Define CSV path of Users and Group ##################

$UserPath = Read-Host -Prompt "Enter File Path For CSV list of users"

$DistributionGroups = Read-Host -Prompt "Enter File Path For CSV list of Distro Groups"

$DistributionGroupMembers = Read-Host -Prompt "Enter File Path For CSV list of Distro Group Members"

################Import CSV list of Users with Passwords and Assign Licenses################################################

Import-Csv -Path $userpath | foreach {New-MsolUser -UserPrincipalName $_.UserPrincipalName -FirstName $_.FirstName -LastName $_.LastName

Import-Csv -Path $DistributionGroups| foreach {New-Distributiongroup -Name $_.Name -PrimarySmtpAddress $_.PrimarySmtpAddress }
```

1. CSV list of All users:
    a. You will want to create a CSV file with the following headers:
        i. UserPrincipalName
        ii. FirstName
        iii. LastName
        iv. DisplayName
        v. Password
        vi. UsageLocation
        vii. LicenseAssignment
            1. This is reseller-account:SPE_E3 for the M365 Buisness Plan. A list of all the license skuIDs can be found here

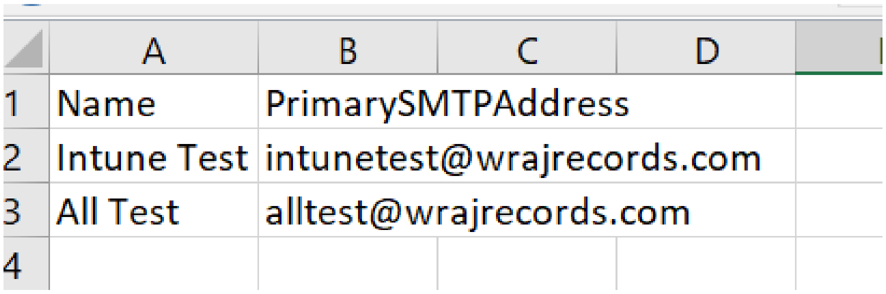
Your CSV should look like the following:



	A	B	C	D	E	F	G	H	I
1	UserPrincipalName	FirstName	LastName	DisplayName	Password	UsageLoca	LicenseAssignment		
2	starfox34@tminus365.com	Star	Fox	Star Fox1	Summ3r1	US	reseller-account:SPE_E3		
3	mjones@tminus365.com	Donkey	Kong	DK1	Matrix@1	US	reseller-account:SPE_E3		
4									
5									

2. CSV list of all Groups you want to create. You will want to include at least one group so you can assign policies that we create to that group.
    a. You will want the CSV file with the following headers
        i. Name
        ii. PrimarySMTPAddress

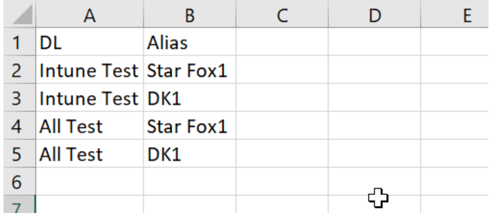
Your CSV Should look like the following:



	A	B	C	D
1	Name	PrimarySMTPAddress		
2	Intune Test	intunetest@wrajrecords.com		
3	All Test	alltest@wrajrecords.com		
4				

3. CSV list of all members part of the Groups you created.
    a. You will want the CSV file with the following headers:
        i. DL
        ii. Alias (Alias is the Display Name of the user)

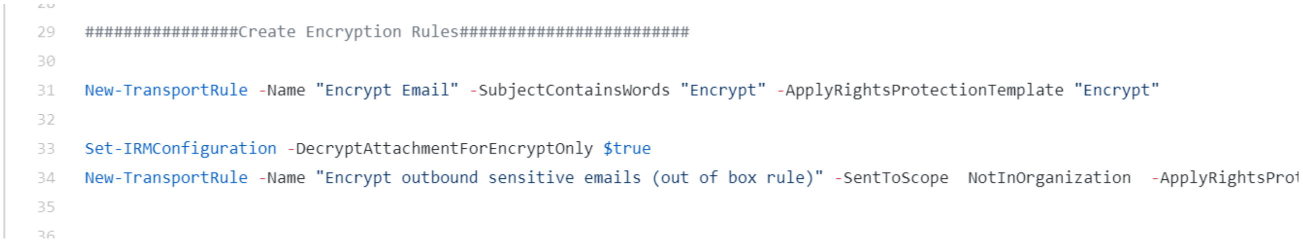
Your CSV should look like the following:



	A	B	C	D	E
1	DL	Alias			
2	Intune Test	Star Fox1			
3	Intune Test	DK1			
4	All Test	Star Fox1			
5	All Test	DK1			
6					
7					

When you enter your file paths you do not have to put them in quotations. Here is an example of my file path: C:\Users\NickRoss\OneDrive - PAX8\Scripts\PowershelltesttemplateM365Biz.csv

Creation of Encryption Mail Flow Rules



```
################Create Encryption Rules########################

New-TransportRule -Name "Encrypt Email" -SubjectContainsWords "Encrypt" -ApplyRightsProtectionTemplate "Encrypt"

Set-IRMConfiguration -DecryptAttachmentForEncryptOnly $true
New-TransportRule -Name "Encrypt outbound sensitive emails (out of box rule)" -SentToScope  NotInOrganization  -ApplyRightsProt
```

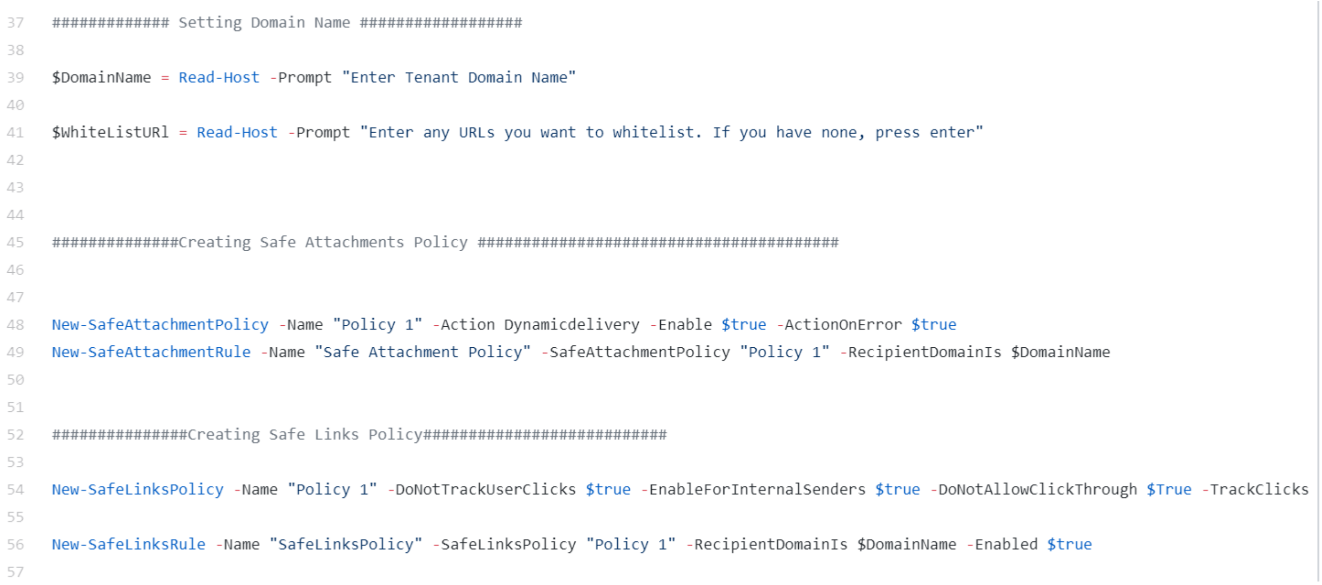
The next section creates 2 mail flow rules:

1. A mail flow rule to encrypt messages with the word “Encrypt” in the subject line
2. A mail flow rule to encrypt messages with the following sensitive data detected:
    a. ABA routing number
    b. Credit card number
    c. Drug Enforcement Agency (DEA) number
    d. U.S./U.K. passport number
    e. U.S. bank account number
    f. U.S. Individual Taxpayer Identification Number (ITIN)
    g. U.S. Social Security Number (SSN)

Note This encrypts the email but not the attachment if the sensitive data is found in an attachment. Once the email is decrypted then the user will be able open the attachment just fine.

Creation of ATP Policies

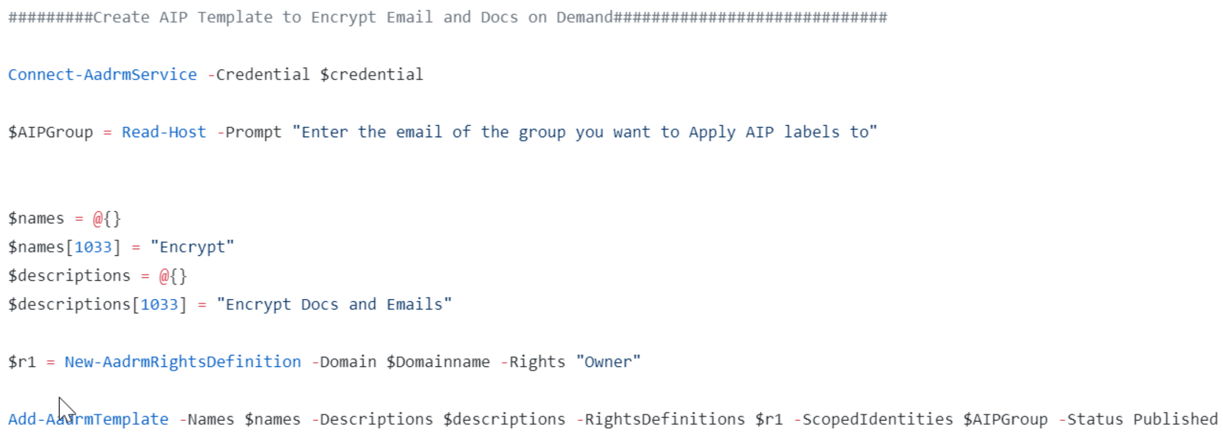
The next section creates a safe links and safe attachment policy in the security and compliance center. By default, these policies are not turned on. This allows you the chance to whitelist urls as well. You will be prompted to enter the domain name of your organization:



```
############# Setting Domain Name ##################

$DomainName = Read-Host -Prompt "Enter Tenant Domain Name"

$WhiteListURl = Read-Host -Prompt "Enter any URLs you want to whitelist. If you have none, press enter"



##############Creating Safe Attachments Policy ########################################


New-SafeAttachmentPolicy -Name "Policy 1" -Action Dynamicdelivery -Enable $true -ActionOnError $true
New-SafeAttachmentRule -Name "Safe Attachment Policy" -SafeAttachmentPolicy "Policy 1" -RecipientDomainIs $DomainName


###############Creating Safe Links Policy###########################

New-SafeLinksPolicy -Name "Policy 1" -DoNotTrackUserClicks $true -EnableForInternalSenders $true -DoNotAllowClickThrough $True -TrackClicks

New-SafeLinksRule -Name "SafeLinksPolicy" -SafeLinksPolicy "Policy 1" -RecipientDomainIs $DomainName -Enabled $true

```

Creation of AIP Label for Encryption

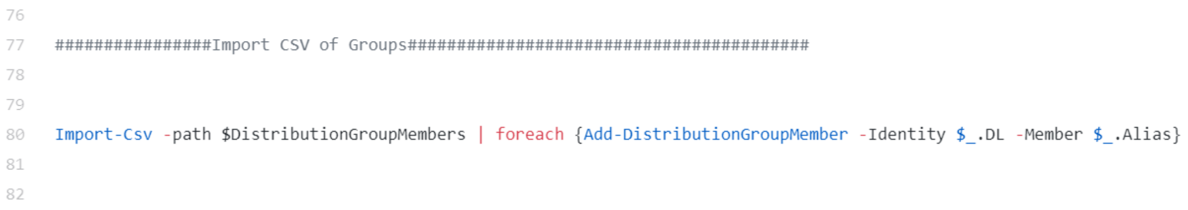
This section creates a new label to apply to a certain group of users that they can use to encrypt documents and emails. If they have installed the Azure Information protection plugin they will be able to apply this label directly. You may want to apply this to a group with all users. It just gives users to encrypt emails and documents on demand. It does not auto-apply



```
#########Create AIP Template to Encrypt Email and Docs on Demand#############################

Connect-AadrmService -Credential $credential

$AIPGroup = Read-Host -Prompt "Enter the email of the group you want to Apply AIP labels to"


$names = @{}
$names[1033] = "Encrypt"
$descriptions = @{}
$descriptions[1033] = "Encrypt Docs and Emails"

$r1 = New-AadrmRightsDefinition -Domain $Domainname -Rights "Owner"

Add-AadrmTemplate -Names $names -Descriptions $descriptions -RightsDefinitions $r1 -ScopedIdentities $AIPGroup -Status Published
```

Adding Members to Groups

This section simply adds members to groups from the CSV file we created. I put this lower to give our groups a chance to propagate. Having them earlier can create errors.



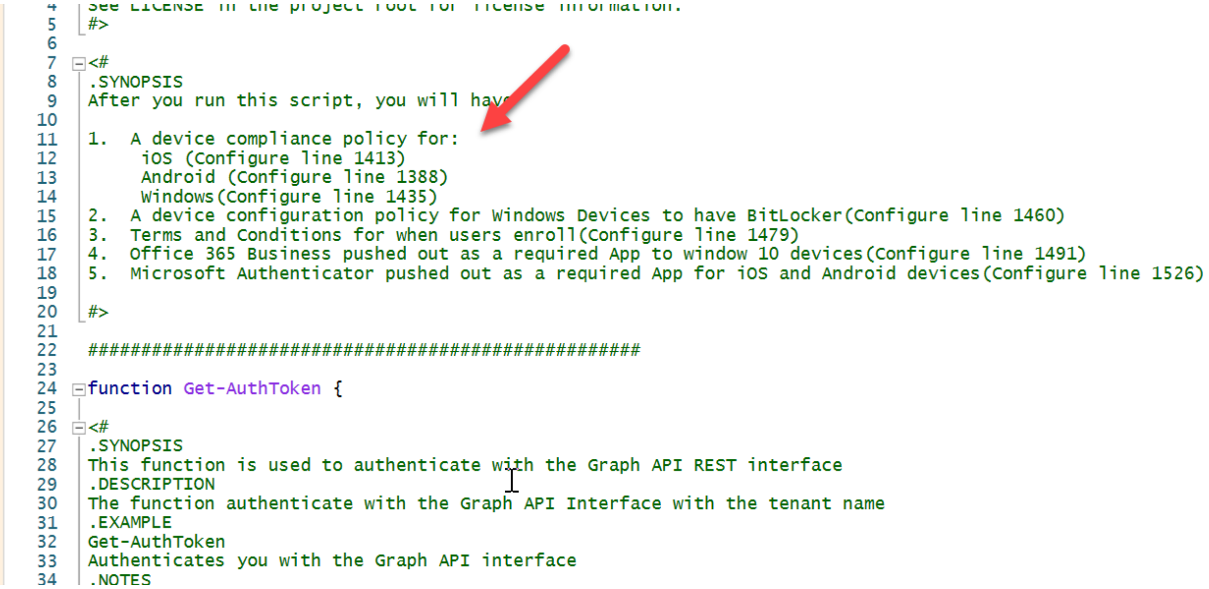
```
################Import CSV of Groups#########################################


Import-Csv -path $DistributionGroupMembers | foreach {Add-DistributionGroupMember -Identity $_.DL -Member $_.Alias}


```

Intune Configuration

Lines 90-1992 focus on the Intune configuration. Here we are connecting to the graph api to configure the following:

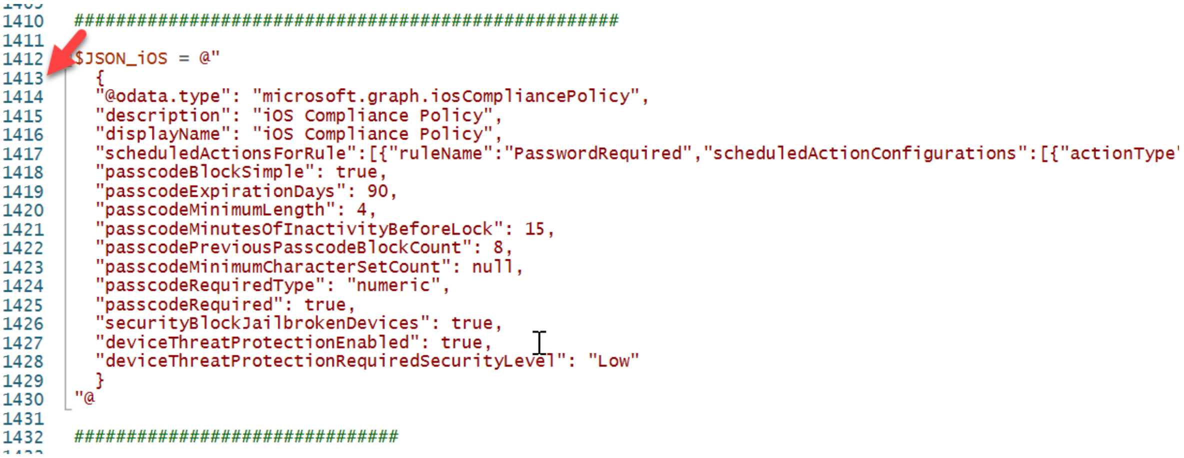
a. A device compliance policy for:
   i. iOS
   ii. Android
   iii. Windows
b. A device configuration policy for Windows Devices to have BitLocker
c. Terms and Conditions for when users enroll
d. Office 365 Business pushed out as a required App to window 10 devices and uninstall existing versions of proplus
e. Microsoft Authenticator pushed out as a required App for iOS and Android devices

1. If you want to modify any of the settings for any of the policies, you can view their respective lines:



```
4    See LICENSE in the project root for license information.
5    #>
6
7    <#
8    .SYNOPSIS
9    After you run this script, you will have
10
11   1.  A device compliance policy for:
12        iOS (Configure line 1413)
13        Android (Configure line 1388)
14        Windows(Configure line 1435)
15   2.  A device configuration policy for Windows Devices to have BitLocker(Configure line 1460)
16   3.  Terms and Conditions for when users enroll(Configure line 1479)
17   4.  Office 365 Business pushed out as a required App to window 10 devices(Configure line 1491)
18   5.  Microsoft Authenticator pushed out as a required App for iOS and Android devices(Configure line 1526)
19
20   #>
21
22   ####################################################
23
24   function Get-AuthToken {
25
26   <#
27   .SYNOPSIS
28   This function is used to authenticate with the Graph API REST interface
29   .DESCRIPTION
30   The function authenticate with the Graph API Interface with the tenant name
31   .EXAMPLE
32   Get-AuthToken
33   Authenticates you with the Graph API interface
34   .NOTES
```

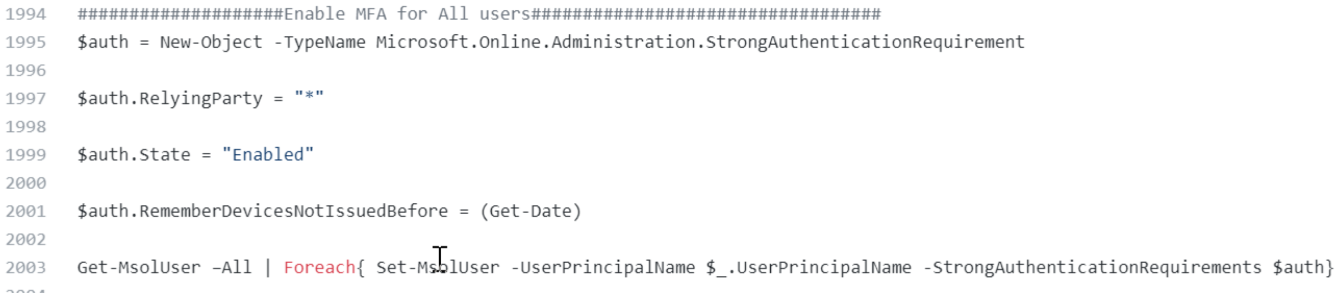
Ex. iOS



```
1410  ####################################################
1411
1412  $JSON_iOS = @"
1413    {
1414    "@odata.type": "microsoft.graph.iosCompliancePolicy",
1415    "description": "iOS Compliance Policy",
1416    "displayName": "iOS Compliance Policy",
1417    "scheduledActionsForRule":[{"ruleName":"PasswordRequired","scheduledActionConfigurations":[{"actionType'
1418    "passcodeBlockSimple": true,
1419    "passcodeExpirationDays": 90,
1420    "passcodeMinimumLength": 4,
1421    "passcodeMinutesOfInactivityBeforeLock": 15,
1422    "passcodePreviousPasscodeBlockCount": 8,
1423    "passcodeMinimumCharacterSetCount": null,
1424    "passcodeRequiredType": "numeric",
1425    "passcodeRequired": true,
1426    "securityBlockJailbrokenDevices": true,
1427    "deviceThreatProtectionEnabled": true,
1428    "deviceThreatProtectionRequiredSecurityLevel": "Low"
1429    }
1430  "@
1431
1432  ###############################
```

Enable MFA for all Users

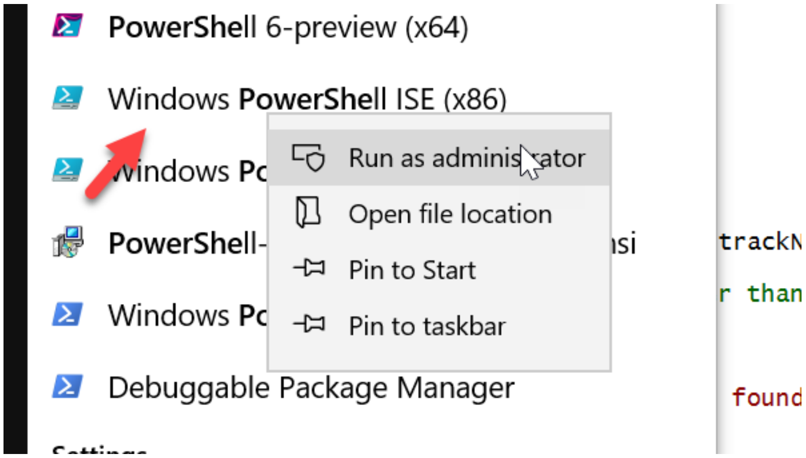
The final settings is enabling MFA for all users. The user will be prompted to add their second factor the next time the login



```
1994  ###################Enable MFA for All users#################################
1995  $auth = New-Object -TypeName Microsoft.Online.Administration.StrongAuthenticationRequirement
1996
1997  $auth.RelyingParty = "*"
1998
1999  $auth.State = "Enabled"
2000
2001  $auth.RememberDevicesNotIssuedBefore = (Get-Date)
2002
2003  Get-MsolUser -All | Foreach{ Set-MsolUser -UserPrincipalName $_.UserPrincipalName -StrongAuthenticationRequirements $auth}
```

Running the Powershell Script

1. Run Powershell ISE as Administrator



2. Copy and paste the script from github

3. Sign-in with the Global admin user

4. Follow the prompts accordingly. Once the script is complete you get a new commandline

Conclusion

I hope this article provided you some targeted guidance on implementing M365 Business with powershell. I hope to get some feedback to improve this guide further. I would also like to hear if there is more content that you would like to see in this guide. Any feedback can be sent to my email below:

Msp4msps@tminus365.com

T-MINUS 365