Table of Contents

[Important Security Note 1](file:///C:\Local\MFAJEA\MFA%20Helpdesk%20-%20User%20Guide.docx#_Toc485141894)

[User Guide 2](#_Toc485141895)

[Connecting to the PowerShell session 2](#_Toc485141896)

[Disconnecting from the PowerShell session 2](#_Toc485141897)

[Setting a MFA bypass for a user 2](#_Toc485141898)

[Removing a MFA bypass for a user 3](#_Toc485141899)

[Getting the list of users with a MFA bypass 3](#_Toc485141900)

[Getting the MFA authentication methods for a user 3](#_Toc485141901)

[Changing the default MFA authentication method for a user 4](#_Toc485141902)

[Getting the aliases of MFA authentication methods 4](#_Toc485141903)

[Resetting the MFA authentication methods for a user 5](#_Toc485141904)

[Changing the cached MSOL credentials 5](#_Toc485141905)

[Logging and Auditing 5](#_Toc485141906)

# Important Security Note

This module requires caching the credentials of an Azure AD Global Admin in JEAHOST. The password of this account is encrypted under the security context of a group managed service account that is used to run the PowerShell remoting session and not visible to the users connected to the session

A knowledgeable administrator of this server can setup a new process that runs under the security context of the group managed service account and decrypt the credentials. Security best practices should be followed to ensure that administrative access to JEAHOST is limited.

See also the topic “[JEA does not protect against admins](https://msdn.microsoft.com/en-us/powershell/jea/security-considerations#jea-does-not-protect-against-admins)” (<https://msdn.microsoft.com/en-us/powershell/jea/security-considerations#jea-does-not-protect-against-admins>) in the PowerShell JEA documentation.

Put this in the balance when evaluating whether to use this module or not.

# User Guide

## Connecting to the PowerShell session

The first step to use the MFA helpdesk functionality is to connect to the PowerShell remote session. This is the session that was created in the Setup Guide in the server [JEAHOST].

To connect to the remote session, use the following command:

**Enter-PSSession -ComputerName [JEAHOST] -ConfigurationName MFAJEA**

All the remaining functions in this guide assume that a remoting session to [JEAHOST] has already been opened.

## Disconnecting from the PowerShell session

It is important to disconnect from the session once the helpdesk tasks are completed.

To disconnect from the remote session, use the following command:

**Exit-PSSession**

There is no need to disconnect and reconnect for each command. Multiple commands can be run inside the same session before disconnecting.

The session will timeout if idle based on the configuration of PowerShell remoting in JEAHOST (default is 2 hours).

## Setting a MFA bypass for a user

To set a MFA bypass for a user run the following command:

**Set-MFABypassForUser -UserPrincipalName <UPN>**

Where <UPN> is the User Principal Name of the user. The command accepts only one <UPN> per call.

By default, the bypass time is set to 15 minutes. Optionally, you can specify the time duration of the bypass by using the parameter BypassTimeInMinutes:

**Set-MFABypassForUser -UserPrincipalName <UPN> -BypassTimeInMinutes <BypassTime>**

The bypass time should be in the range of 15 minutes to 24 hours. Providing a value for ByPassTimeInMinutes lower than the minimum will use the minimum allowed. Providing a value higher than the maximum will use the maximum allowed.

No new bypass can be set for users that already have a bypass set. If a bypass is tried for a user that already has one then the command will report an error indicating that there is already a bypass with the time remaining for it and not create a new one.

NOTE: the MFA bypass is granted via changes to the membership of a group that exists in the same domain as the ADFS servers. The bypass will then start to take effect once the domain controllers in the same domain as ADFS and that are being used by the ADFS servers (usually those in the same site as the ADFS servers) receive the replication of the group membership. The JEAHOST is setup in the same site as ADFS to minimize this time. Take this replication time into consideration when calculating how much time the users have to wait before the MFA bypass takes effect, especially in the case that the ADFS servers are in multiple AD sites. Note that this replication time is between domain controllers in the same domain as ADFS that the ADFS servers are actively using and not the replication time to all the domain controllers in that domain.

Also, granting a bypass will take effect only the next time that the user is authenticated by ADFS. This can be achieved by closing and reopening the browser or application that is requesting MFA. In the case that the user has a PSSO token (or has KMSI) then it might require clearing the browser cookies before the bypass takes effect.

Take these two things into consideration when training your helpdesk.

## Removing a MFA bypass for a user

To remove a MFA bypass for a user run the following command:

**Remove-MFABypassForUser -UserPrincipalName <UPN>**

Where <UPN> is the User Principal Name of the user. The command accepts only one <UPN> per call.

The command will show an error if the user has no MFA bypass set.

NOTE: removing the MFA bypass will not invalidate the tokens that the user already has issued by ADFS or Azure AD. Removing the MFA bypass only takes effect the next time that the user authenticates to ADFS. This generally happens after the user closes and reopens the browser.  
Keep this in mind in the case that you have KMSI or PSSO in ADFS and/or Azure AD.

## Getting the list of users with a MFA bypass

To get the list of users set for MFA bypass run the following command:

**Get-MFABypassList**

The command will output a table with each row containing the user principal name and the MFA bypass time remaining (in seconds).

## Getting the MFA authentication methods for a user

Run the following command to get the MFA authentication methods for a user:

**Get-MFAUserAuthenticationMethods -UserPrincipalName <UPN>**

Where <UPN> is the User Principal Name of the user. The command accepts only one <UPN> per call.

The output will show the authentication methods registered for the user. Methods not registered are not listed.

The methods are shown using the aliases described in Table 1.

A second column named IsDefault is added to the output. The column will contain the value “True” for the default authentication method for the user. All the other registered methods will show a value of “False” in this column.

## Changing the default MFA authentication method for a user

Run the following command to change the default MFA authentication method for a user:

**Set-MFAUserDefaultAuthenticationMethod -UserPrincipalName <UPN> -Method <MethodAlias>**

Where <UPN> is the User Principal Name of the user. The command accepts only one <UPN> per call, and <Method> is alias of MFA authentication method to set as default for the user.

The list of aliases and their meaning are listed in Table 1.

The user should have already registered this method for the change to take effect.

An error is shown if the user doesn’t have the selected method registered.

NOTE: it might take a couple of minutes before the user the default authentication methods changes takes effect in Azure MFA.

## Getting the aliases of MFA authentication methods

Run the following command to get the aliases of the MFA authentication methods:

**Show-MFAMethodsNames**

The output will show the same aliases as the following table:

|  |  |
| --- | --- |
| Alias | Description |
| TwoWayVoiceMobile | Phone call to authentication phone |
| TwoWayVoiceAlternateMobile | Phone call to alternate authentication phone |
| TwoWayVoiceOffice | Phone call to office phone |
| OneWaySMS | One-time password sent via SMS message to authentication phone |
| PhoneAppOTP | One-time password generated from the Authenticator app |
| PhoneAppNotification | Push notification to the Authenticator app |

Table 1 - MFA Methods Aliases and Descriptions

## Resetting the MFA authentication methods for a user

Run the following command to reset the MFA authentication methods for a user:

**Reset-MFAUserAuthenticationMethods -UserPrincipalName <UPN>**

Where <UPN> is the User Principal Name of the user. The command accepts only one <UPN> per call.

The effect of running this command is that the user will have to register again for Azure MFA.

NOTE: it might take a couple of minutes before the user can register again the authentication methods in Azure MFA.

## Changing the cached MSOL credentials

In the case that the account used to connect to Azure AD changes its password, or there is a need to use a different account by the module, run the following command:

**Initialize-MFACredentialsCache**

And enter the user name and password for the account to be used to connect to Azure AD.

NOTE: the account should be a Global Admin and should not require MFA.

NOTE2: only users that are also local administrators in the JEA host can run this command. Normal helpdesk users (those that are not local administrators) can no run this command.

# 

# Logging and Auditing

The PowerShell remoting session is configured to create a transcript file inside the folder c:\mfahelpdesk\transcripts. Each new session will create a new transcript file in that folder.

The content of the transcript file includes the start time and end time of the session, the user that connected to the session and the commands run by that user, plus other information.

It is important to periodically backup and review these files for auditing.

More information about PowerShell JEA logging and transcripts can be found in: <https://msdn.microsoft.com/en-us/powershell/jea/audit-and-report>