SECUREAUTH FOR OFFICE 365
# TABLE OF CONTENTS

**Introduction** ................................................................. 3

**Best Practices** ................................................................. 4
  - Enterprise Security Concerns .................................................. 5
  - End-User Experience Considerations ......................................... 6
  - MFA Security Considerations .................................................. 6
  - Office 365 Authentication Strategies .......................................... 7
  - Implementing Best Practices for Office 365 ......................... 7
  - Implementing Adaptive Authentication in Front of Office 365 .... 8

**Risks and Considerations: WS-Trust vs. WS-Fed** ..................... 9

**SSO Configuration Guidelines** ........................................... 13
  - Step 1: Prepare for SSO ..................................................... 15
  - Step 2: Prepare Active Directory ........................................... 15
  - Step 3: Create the Required Realms ....................................... 15
  - Step 4: Prepare Office 365 .................................................. 16
  - Step 5: Verify the SSO with IdP ........................................... 20

**Conclusion** ..................................................................... 21
Introduction

Office 365 Security Concerns

Office 365 has become the most widely used cloud service in the world — which makes it a prime attack vector. According to the 2017 Microsoft Security Intelligence Report, Microsoft cloud user accounts saw a 300% increase in cyberattacks over the past year. The report found the majority of these compromises were the result of weak, guessable passwords and poor password management, followed by targeted phishing attacks and breaches of third-party services. Also, on the rise are account sign-ins attempted from malicious IP address – up 44% in 2017.

Many applications and Microsoft Outlook clients use the WS-Trust protocol, including the mail apps that come with iPhones and Macs. The problem with this legacy protocol is that it doesn't allow two-factor authentication (2FA) and only supports password-based authentication. With the amount of sensitive information flowing through your Office 365 instance, password only protection is simply inadequate.

Since attackers are continually finding new ways to defeat authentication challenges, including those based on 2FA, organizations must constantly adapt to defend themselves.
Why SecureAuth?

SecureAuth IdPs adaptive access management for Office 365 provides the greatest protection against credential-based attacks while providing the best user experience. Multiple pre-authentication risk checks coupled with multi-factor authentication (MFA) provide a layered defense nearly impossible to penetrate, even with compromised credentials.

Additionally, the threat data collected via the multiple risk checks can be shared with the SIEM or SOC for correlation with other threat data to help pinpoint threats in a sea of alerts and potential problems.

Instead of interrupting users for an MFA step at every access request, SecureAuth adaptive authentication enables access for low-risk requests without an MFA step, requires MFA for medium-risk attempts, and denies or redirects high-risk request.

SecureAuth adaptive access control ensures strong security without sacrificing usability. In addition, setting up adaptive authentication for Office 365 in SecureAuth IdP is as simple as designating the appropriate options in several drop-down boxes. Then you define the rules and customize the response of SecureAuth IdP to your preferred use cases. A member of the Customer Success team will be happy to help streamline this process.

About this Document

This document explores the best methods for securing Office 365 using SecureAuth IdP and will help you determine which techniques offer the best combination of security and efficiency for your organization.

Best Practices

In order to successfully implement SecureAuth IdP with Office 365, the following questions should be considered:

+ How will implementing Office 365 behind SecureAuth IdP increase my enterprise security?
+ What kind of user experience will be encountered?
+ What are the different MFA options SecureAuth IdP can use to authenticate users into Office 365?
+ What authentication options would I like to implement with Office 365?
+ What are our best practices and recommendations?
+ How will adaptive authentication factor into Office 365 workflows?

Each of these questions is addressed on the following pages.
Enterprise Security Concerns

Three questions are essential to building your Office 365 security strategy, and your enterprise strategy overall:

+ Which applications within Office 365 are you trying to leverage?
+ Which client devices will you use to access those applications: Macs using MacOS, PCs using Windows, or mobile devices using iOS, Android, Windows, or some other OS?
+ What are your use cases?

When determining the level of security needed, consider with which types of devices users will access Office 365 and which Office 365 applications (Word, Excel, PowerPoint, OneDrive) will they access on these devices? Once you have answered these questions, you can begin to plan the best methods for protecting those different access paths and determine your user cases.

For example, if you are using a Windows PC inside the network, and are local to the domain, do you want to use integrated Windows Authentication to allow the user to seamlessly gain access or some other authentication protocol? Whichever way you choose, you will need to create an adaptive rule that stipulates that if you are on this internal subnet, or if you are not on this internal subnet, you must perform this workflow. From there, you can begin to prescribe each use case.

You must make the decision early about which applications you will allow and how you will enable that access. For instance, you may not want to allow the native client from an iOS device (such as Mail) or an Android device bundled with the OS to be accessed or even allowed. Are these devices and the applications on them to be publicly accessible on a home PC or is this a private device (BYOD)? Are they only available on a mobile device?

The key to the successful roll-out of a cloud or SaaS application is understanding how you are going to be accessing it. Until you understand these questions and have identified your use cases, you cannot start setting up security.

SecureAuth IdP’s adaptive authentication is the foundation of a sound security policy. Adaptive authentication is an additional safety net that doesn’t inhibit the daily routine of users — and is nearly impossible for attackers to get past. Plus, the threat data collected during the multiple risk checks can be shared with your SIEM or SOC for correlation with other threat data, helping you pinpoint identity-based threats in a sea of alerts and potential problems. For more information on the adaptive authentication capabilities offered by SecureAuth IdP, refer to the Adaptive Authentication Best Practices Guide.
End-User Experience Considerations

Office 365 supports three protocols today:

+ WS-Federation
+ WS-Trust
+ SAML

SecureAuth recommends using WS-Fed (you might see the term WS-* or WS security since there is more than just federation included in this security protocol) instead of WS-Trust or SAML. For single sign-on (SSO), SecureAuth recommends WS-Fed because it is Microsoft’s native federation framework. This protocol provides the greatest level of support for Office 365, since it includes not just web-based authentication (such as supporting simple logins), but also supports users who are accessing the network via thick clients that may not use web-based authentication.

SAML can only be used with Office 365 through the intermediary of ADFS and is therefore an unnecessary third step, unless there is some valid reason for its use.


For more information on the prerequisites, requirements, and configuration guidelines for using SSO with Office 365 and SecureAuth, refer to “SSO Configuration Guidelines” starting on page 13.

MFA Security Considerations

When considering an MFA configuration, the rule of thumb is to balance strong security with the best end-user experience. If you have users who are using a native app on their device (such as Exchange on Windows), you don’t want them to get a push notification on that same device to authenticate. However, consider the convenience of the user as well. If the device is secure, an extra layer of security might represent an extra level of interruption that hinders the overall end-user experience.

SecureAuth IdP provides the ability to use many multi-factor methods (including phone, time-based passcode and email-based methods) that are secure.

In general, your mobile app interactions represent your most secure methods of authentication.
Office 365 Authentication Strategies

Office 365 uses three strategies for authentication (excluding available Microsoft add-ons):

+ All-Cloud Managed Identities
+ Microsoft Directory Sync tool
+ Federation (recommended)

All-Cloud Managed identities reside strictly within Office 365 and the Azure Active Directory; therefore, you are logging in ‘locally’ to Office 365.

The Directory Sync tool synchronizes passwords, so you are still gaining SSO, a single set of credentials to log in, both locally to your own domain and remotely to the Microsoft cloud. However, this requires you to send password hashes from your local AD up to the cloud. For security reasons, we believe this is a more problematic approach. In general, you should avoid sending your user password outside of your organization, even in a hashed form.

Federation does not require password syncing – that is syncing your AD to the cloud – but does require attribute syncing. By using cloud identifiers or mutable IDs, which are objects that link the local AD to the cloud Office 365 AD, your IdP (such as ADFS or SecureAuth) can then assert that information to Office 365 to authenticate the user when required. For many reasons, Federation is the recommended method, since it is both the strongest and most secure way to authenticate.

If you are using ADFS or if you cannot use MFA natively (and must use a third-party product with fewer MFA methods available) you may not derive sufficient benefit from federation and should look to All-Cloud or Directory Sync for your solution.

Implementing Best Practices for Office 365

Follow the Microsoft Best Practices for Office 365. This includes the following topics:

+ Using MFA
+ Advanced Security Management (ASM)
+ Secure Mail Flow
+ Mailbox Audit Logging
+ Configuring Data Loss Prevention (DLP)
+ Customer Lockbox
+ Using Office 365 Secure Score

Microsoft is continually updating Office 365, so always check with the Microsoft website to make sure you are conversant with the latest updates.

Here is a general rule of thumb for Office 365 users wanting to know whether SecureAuth supports a particular feature: if ADFS can support it, so can SecureAuth IdP, and normally at a more comprehensive level.
Implementing Adaptive Authentication in Front of Office 365

Setting up adaptive authentication for Office 365 in SecureAuth IdP is as simple as designating the appropriate options in several drop-down boxes. Once you’ve placed SecureAuth in front of Office 365 with Federation, you define the rules and customize the response of SecureAuth IdP to your preferred use case(s). We recommend working with a SecureAuth Solutions Architect to streamline this process. Make sure you stipulate the use cases you require, as well as the applications, platforms, and devices that will be used in performing the authentication, and identify the end users who will be using Office 365. Your SecureAuth Solutions Engineer can help in defining these goals.

Once you have decided on these parameters, make your SecureAuth Solutions Architect aware of them. At a minimum, you should plan to enable SecureAuth Threat Services, particularly on a WS-Trust endpoint -- a service that only SecureAuth can provide today. Correctly defining and interrogating a WS-Trust endpoint can determine whether a request is authentic or is coming from an unknown and unreliable source. To increase security, also enable geo-location awareness around the endpoint to further increase threat analysis. In addition, make sure to separate and define your internal network so that it is effectively isolated from the external network.
Risks and Considerations: WS-Trust vs. WS-Fed

Microsoft Office 365 (Office 365) primarily leverages two protocols to authenticate users: WS-Trust and WS-Fed.

In the context of Office 365, WS-Trust is a legacy protocol that is used by older Microsoft applications and by non-Microsoft client applications.

Microsoft often regards the WS-Fed protocol as “modern authentication”. This authentication method evokes an embedded browser within native applications, and uses web flows to authenticate users before allowing them to use the application. As its name implies, WS-Fed is a more modern authentication method and therefore more secure than the legacy WS-Trust protocol. SecureAuth provides strong multi-factor and adaptive authentication for this protocol.

WS-Trust only allows for user ID and password authentication without an ability to execute Two-Factor Authentication (2FA). In the absence of modern authentication, SecureAuth provides the best identity security for applications using WS-Trust.

We recommend using modern authentication exclusively whenever possible.

SecureAuth has monitored and detected many methodical attacks against Office 365 using the WS-Trust protocol. Office 365 is a common attack target since it is inherently weak (ID/Password only) and uses a transparent service-based login method (not a website).
Using increasingly sophisticated techniques, attackers can now penetrate WS-Trust-protected applications containing user data such as SharePoint and OneDrive. These attacks appear to be targeting Office 365 customers worldwide, including customers using Microsoft ADFS, and likely any WS-Trust token validation product.

Most WS-Trust traffic is proxied by Microsoft before being forwarded to the IdP for validation. Note that the IdP appliance can either be inside the DMZ or within the network. Applications such as Skype for Business make a direct call to IdP, bypassing the Office 365 cloud proxy.

The challenge to moving to WS-Fed exclusively is to determine whether your business requirements allow you to disable WS-Trust. This process involves identifying applications that use WS-Trust in your enterprise. It is not just old Office apps that use WS-Trust; many current applications, such as mail apps supplied with iPhone and Mac, still use WS-Trust.

Table 1 and Table 2 illustrate a few, but not all, client types that use WS-Trust. For example, there are third-party Mobile Device Management (MDM) apps as well as VoIP apps that only use WS-Trust.

<table>
<thead>
<tr>
<th>Client Application</th>
<th>Supported Protocol</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office 2016 (Windows)</td>
<td>WS-Fed</td>
<td></td>
</tr>
<tr>
<td>Office 2013 (Windows)</td>
<td>WS-Fed or WS-Trust</td>
<td>WS-Fed requires patch and registry settings</td>
</tr>
<tr>
<td>Office 2010 (Windows)</td>
<td>WS-Trust</td>
<td></td>
</tr>
<tr>
<td>Office 2016 (Mac)</td>
<td>WS-Fed with update</td>
<td></td>
</tr>
<tr>
<td>Office 2011 (Mac)</td>
<td>WS-Trust</td>
<td></td>
</tr>
<tr>
<td>Office/Outlook (iOS)</td>
<td>WS-Fed with update</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Supported Protocols for Office Versions
### Table 1. Supported Protocols for Office Versions (continued)

<table>
<thead>
<tr>
<th>Client Application</th>
<th>Supported Protocol</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office/Outlook (Android)</td>
<td>WS-Fed with update</td>
<td></td>
</tr>
<tr>
<td>Office/Outlook (Windows Phone)</td>
<td>See table below</td>
<td></td>
</tr>
<tr>
<td>Other office apps</td>
<td>See table below</td>
<td></td>
</tr>
<tr>
<td>Native iMAP clients</td>
<td>WS-Trust</td>
<td>Example: Native iOS, Mac, Android mail, and calendar apps</td>
</tr>
</tbody>
</table>

### Table 2. Office Client vs. Platform Cross reference

<table>
<thead>
<tr>
<th>Office Client Application</th>
<th>Windows</th>
<th>Mac OS X</th>
<th>Windows Phone</th>
<th>iOS</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office clients</td>
<td>Available now for Office 365 and Office 2016</td>
<td>Available now for Office 2016 Also available for OneNote 2014</td>
<td>Available now</td>
<td>Word, Excel, and PowerPoint are available for phones and tablets</td>
<td>Word, Excel and PowerPoint are available now for both phones and tablets</td>
</tr>
<tr>
<td>Skype for Business (formerly Lync)</td>
<td>Included in Office client</td>
<td>Available now</td>
<td>Available now</td>
<td>Available now</td>
<td>Available now</td>
</tr>
<tr>
<td>Outlook</td>
<td>Included in Office client</td>
<td>Available now</td>
<td>Coming soon</td>
<td>Available now</td>
<td>Available now</td>
</tr>
<tr>
<td>OneDrive for business</td>
<td>Included in Office client</td>
<td>Available now</td>
<td>Available now</td>
<td>OneDrive for Business is available now</td>
<td>OneDrive for Business is available now</td>
</tr>
<tr>
<td>Legacy clients</td>
<td>There are no plans for Office 2010 or Office 2007 to support ADAL-based authentication</td>
<td>There are no plans for Office for Mac 2011 to support ADAL-based authentication</td>
<td>There are no plans for Office on Windows Phone 7 to support ADAP-based authentication</td>
<td>There are no plans to enable older Outlook iOS clients</td>
<td>There are no plans to enable older Outlook Android clients</td>
</tr>
</tbody>
</table>

For more information on these versions and their support, refer to this [Microsoft.com article on modern authentication](https://microsoft.com/article-on-modern-authentication).
If your organization needs to use WS-Trust, SecureAuth recommends implementing the SecureAuth blocking features to limit the client application types that can access the service, thereby greatly reducing the attack surface.

**NOTE:** If you are using a version of SecureAuth IdP prior to 9.1, please contact your SecureAuth account representative for further details on how we can help you.

The SecureAuth WS-Trust Blocking feature enables you to either whitelist or blacklist WS-Trust validation requests based on:

**IP Address**

![Figure 2. IP Address Blocking Feature 1](image)

**User Agent**

![Figure 3. User Agent Blocking](image)

**Application Type**

![Figure 4. Application Blocking](image)

For example, if the use case requires some users to use the native iOS or Mac (non-Microsoft) email application, a filter can be enabled to only allow the specific client application types.

In summary, WS-Trust is a legacy protocol that can be reasonably protected, leveraging the SecureAuth adaptive features, but due to the architecture of the protocol itself, it is not as robust as WS-Fed. As a result, SecureAuth recommends using WS-Fed exclusively, or using WS-Trust in a limited capacity to reduce the attack surface.
SSO Configuration Guidelines

This section provides detailed information about how to enable Adaptive Authentication for Windows Office 365 external users and single sign-on (SSO) for internal users by integrating SecureAuth IdP to access Office 365 web and thick applications via WS-Trust and WS-Federation.

NOTE: This section provides a more detailed look into the complexities that are often encountered in an SSO configuration of SecureAuth IdP with Office 365. Though a straightforward process, we highly recommend working closely with a SecureAuth Solutions Engineer to achieve the best possible results.

SSO enables users to access Microsoft Cloud services using your organization's Active Directory corporate credentials. Figure 5 depicts a flowchart of this process.

Identity Synchronization and Federation

SecureAuth IdP enables companies to place SecureAuth Adaptive Authentication component on top of the standard authentication options to verify users trying to access an Office 365 site are authorized to use company resources.

Figure 5. Identity Synchronization and Federation Flowchart
An example of this is illustrated in figure 6.

![Diagram of advanced authentication](image)

**Figure 6.** Advanced Authentication

To integrate Office 365 to SecureAuth IdP to implement SSO, take the following steps:

1. Prepare for SSO.
2. Prepare Active Directory and set up your IdP.
3. Create IdP realms.
4. Set up directory synchronization.
5. Verify SSO with SecureAuth IdP.

Each of these steps is detailed in the following subsections.
Step 1: Prepare for SSO

The first step is to review your requirements for implementing SSO. Prerequisites include:

+ Microsoft Azure Active Directory Module for Windows PowerShell set up to establish a federated trust between your on-premises IdP and Azure AD.
+ Required Microsoft cloud service subscription updates installed to ensure your users are running the latest versions of Windows 7, Windows Vista, or Windows XP. Some features may not function properly without the appropriate versions of operating systems, browsers, and software.

Step 2: Prepare Active Directory

To prepare Active Directory for SSO:

+ Configure necessary Active Directory settings.
+ The user principal name (UPN), also known as a user log-on name, must be set up in a specific way for each user.
+ Run the Microsoft Deployment Readiness Tool. This tool inspects your Active Directory environment and provides a report including whether or not you are ready to set up SSO. If not, it lists the changes you need to make to prepare for SSO. For example, it inspects whether or not your users have UPNs and if those UPNs are in the correct format.
+ Depending on your domain setup, the following adjustments may need to be made:
  - The UPN must be set and known by the user.
  - The UPN domain suffix must be under the domain to be used for SSO.
  - The domain to be federated must be registered as a public domain with a domain registrar or within your own public DNS servers.
  - To create UPNs, follow the instructions in the Active Directory topic, Add User Principal Name Suffixes. Note that UPNs used for SSO can only contain letters, numbers, periods, dashes, and underscores.
  - If your Active Directory domain name is not a public Internet domain (for example, it ends with a ".local" suffix), you must set a UPN to have a domain suffix that is under an Internet domain name that can be registered publicly. We recommend that you use something familiar to your users, such as their email domain.
  - If you have already set up Active Directory synchronization, the user's UPN may not match the user's on-premises UPN defined in Active Directory. Resolve this by renaming the user's UPN using the Set-MsolUserPrincipalName cmdlet in the Microsoft Azure Active Directory Module for Windows PowerShell.

Step 3: Create the Required Realms

Follow the steps contained in SecureAuth IdP Realm Guide to create one or more required realms.
Step 4: Prepare Office 365

To prepare Office 365 for synchronization and deployment with IdP:

1. Review Office 365 Domain Registration Prerequisites

The following domain registration prerequisites must be met before deploying Office 365 for SecureAuth SSO.

+ Onmicrosoft.com admin account
+ Add a domain
+ Configure DNS
+ Leave the onmicrosoft.com domain as the primary domain for the account since any new domain causes errors when using the `Set-MsolDomainAuthentication` command in PowerShell.

2. Deploy MSOnline PowerShell for Azure Active Directory

Use MSOnline PowerShell for Azure Active Directory to configure Office 365 for SecureAuth SSO. To deploy this tool:

+ Find MSOnline PowerShell for Azure Active Directory at this link.
+ When using PowerShell, inspect the directory in which MSOnline PowerShell will be deployed by typing:
  ```powershell
  PS> Save-Module -Name MSOnline -Path <path>
  SecureAuth IdP has local policies that may block the Secure connection from PowerShell to MSOL.
  ```
  ```powershell
  PS> Install - Module AzureAD Preview
  This requires PS 3.0 or greater as well as .NET 4.5 or greater.
  ```

3. Connect Active Directory with Azure Active Directory

Azure AD Connect will integrate your on-premises directories with Azure Active Directory. This enables you to provide a common identity for your users on Office 365. Follow these steps to set up functionality:

+ Download Azure Active Directory Connect (click here to download) and install using Express settings
+ Perform an Azure AD Connect – configure filtering and implement password synchronization
+ Install the Microsoft Online Services Sign-in Assistant for IT Professionals

+ Install the Azure Active Directory for PowerShell on the computer desktop.
4. Configure Office 365 Domain Federation via PowerShell

+ Click [here](#) to download and install the Azure Active Directory for Windows PowerShell executable.

+ Open the PowerShell from desktop by right-clicking and specifying Run as Administrator.

**NOTE:**

+ Azure Active Directory PowerShell does not always behave properly with the SecureAuth IdP appliance.

+ Download and open this version of PS on a non-appliance computer using the same domain and subnet. This will still federate the domain.

+ The cmdlet `Connect -MsolService` initiates the connection to the online service.

+ The preceding steps will initiate the various endpoints that are necessary to connect SSO.

+ Make sure there are no spaces or line breaks anywhere in the command list where they do not belong.

+ At the PS command line, enter this text:

  ```
  PS> Connect -MsolService
  ```
This command begins the process of federating the connection. Other commands to enter in PS include:

<table>
<thead>
<tr>
<th>PS Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ura=&quot;<a href="https://SecureAuthIdPFQDN/SecureAuthIdPRealm2/webservice/wstrust.svc/2005/usernamemixed">https://SecureAuthIdPFQDN/SecureAuthIdPRealm2/webservice/wstrust.svc/2005/usernamemixed</a>&quot;</td>
<td>The variable defines the SecureAuth IdP FQDN and Office 365 Realm 2, followed by /webservice/wstrust.svc/2005/usernamemixed. This URL specifies the endpoint used by active clients when authenticating with domains set up for SSO (identity federation) in Office 365. For example: &quot;<a href="https://secureauth.company.com/secureauth2/webservice/wstrust.svc/2005/usernamemixed">https://secureauth.company.com/secureauth2/webservice/wstrust.svc/2005/usernamemixed</a>&quot;. Note: SecureAuth IdP FQDN and SecureAuth IdP Realm2 are unique for every appliance on which you have created two or more realms.</td>
</tr>
<tr>
<td>$url=&quot;<a href="https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/">https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/</a>&quot;</td>
<td>The variable defines the SecureAuth IdP FQDN and Office 365 Realm 1. This URL is where web-based clients are directed when signing into Office 365. For example: &quot;<a href="https://secureauth.company.com/secureauth1/">https://secureauth.company.com/secureauth1/</a>&quot;</td>
</tr>
<tr>
<td>$uri=&quot;<a href="https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/">https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/</a>&quot;</td>
<td>The variable defines the SecureAuth IdP FQDN and Office 365 Realm 1. This is the unique identifier of the domain in the Office 365 platform that is derived from the federation server. For example: &quot;<a href="https://secureauth.company.com/secureauth1/">https://secureauth.company.com/secureauth1/</a>&quot;. Note: The uri command and the WSFed/SAML Issuer in the SecureAuth IdP Web Admin must match exactly, including the trailing forward slash &quot;.&quot;.</td>
</tr>
<tr>
<td>$logouturl=&quot;<a href="https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/wsfsdsignout.aspx">https://SecureAuthIdPFQDN/SecureAuthIdPRealm1/wsfsdsignout.aspx</a>&quot;</td>
<td>The variable defines the SecureAuth IdP FQDN and Office 365 Realm 1, followed by /wsfsdsignout.aspx. This is the URL to which users are redirected for signing out of Office 365. If using both IdP-initiated and SSO, and experience issues logging in, then contact Support. For example: &quot;<a href="https://secureauth.company.com/secureauth1/wsfsdsignout.aspx">https://secureauth.company.com/secureauth1/wsfsdsignout.aspx</a>&quot;</td>
</tr>
</tbody>
</table>

Table 4. PS Commands to Federate Office 365 with SecureAuth IdP
<table>
<thead>
<tr>
<th>PS Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$metadata=&quot;<a href="https://SecureAuthIdPFQDN/SecureAuthIdPRealm2/webservice/wstrust.svc/mex">https://SecureAuthIdPFQDN/SecureAuthIdPRealm2/webservice/wstrust.svc/mex</a>&quot;</td>
<td>The variable defines the SecureAuth IdP FQDN and Office 365 Realm 2, followed by the metadata location, such as /webservice/wstrust.svc/mex. This URL specifies the metadata exchange endpoint used for authentication from rich client applications, such as Lync Online. Example: <a href="https://secureauth.company.com/secureauth2/webservice/wstrust.svc/mex">https://secureauth.company.com/secureauth2/webservice/wstrust.svc/mex</a>&quot;</td>
</tr>
<tr>
<td>$cert=&quot;&lt;CERT VALUE&gt;&quot;</td>
<td>The variable defines the Certificate Value of the certificate used to sign tokens passed to the Office 365 identity platform. Replace &lt;CERT VALUE&gt; with the actual value (cert MUST be a single line with no breaks or spaces). Export the certificate used in the SecureAuth IdP Web Admin for signing the WS-Federation Assertion in this way: 1. Export the SSL certificate in Base64 format. 2. Open the exported certificate in a text editor (Windows Notepad or Notepad++). 3. Remove the Begin Certificate and End Certificate lines from the file. 4. Remove all returns (CR-LF) so that the certificate value is one line of text with no formatting.</td>
</tr>
</tbody>
</table>

Table 4. PS Commands to Federate Office 365 with SecureAuth IdP
+ Run these commands exactly in the order provided where: DomainName designates SecureAuth IdP Domain Name SecureAuthIdPFQDN designates the actual SecureAuth IdP Hostname

SecureAuthIdPRealm1 and SecureAuthIdPRealm2

Designate the actual SecureAuth IdP realms being used (in this case, SecureAuth1 and SecureAuth2).

+ Run these commands in a single line; this configures Office 365 with the variables set in the previous lines:

```
```

+ Place quotation marks around the links used. That is, if the command requires

$dom="DomainName", enter the domain name in quotes ($dom="idpXX.secureauth-training.com").

+ Verify that the Office 365 account is configured properly by entering this command into Azure PowerShell:

```
Get-MsolDomainFederationSettings -DomainName <DomainName>
```

in which <DomainName> is the actual domain name, as in this example:

```
Get-MsolDomainFederationSettings -DomainName idpXX.secureauthtraining.com
```

Once these steps have been completed, Office 365 login will automatically redirect to SecureAuth for authentication. In the event of faulty federation, use your administrative onmicrosoft.com login. De-federate your tenant to prevent irreversible changes that cause federation to function improperly.

**Step 5: Verify the SSO with IdP**

Test the Login SSO Office 365 as follows:

1. Open a browser and enter:

https://portal.office.com

2. Begin typing any email ending in your federated domain using this format:

someemail@yourdomain.com

3. You should be redirected to the SecureAuth portal for log on.
Conclusion

The most efficient integration of Office 365 with SecureAuth IdP requires careful preparation with an eye to understanding the organization's use case(s). An organization should strive for a balance between strong security and the best end-user experience.

Finally, a recommendation: as shown in the section SSO Configuration Guidelines, the procedure required to integrate Office 365 with SecureAuth IdP (or any other security software) requires an expert, such as a SecureAuth Solutions Engineer, to ensure a smooth and pain-free deployment. Contact Customer Success team to get the resources you need – we are here to support you.