O365 SSO Identity Provider Installation Guide

SurePassID Authentication Server 23.1

SurePassID, Corp.
360 Central Avenue
First Central Tower
Suite 800
St. Petersburg, FL 33701
USA
+1 (888) 200-8144
www.surepassid.com

Table of Contents

About the SurePassID O365 SSO Identity Provider

This guide explains how install and configure the SurePassID O365 SSO Identity Provider. The SurePassID O365 SSO Identity Provider allows your company to add MFA security to all your Office 365 applications such as Outlook, Word, SharePoint, Excel, etc.

This guide provides information on the following topics:

- **What is SurePassID O365 SSO Identity Provider?**
  - A brief introduction to the O365 SSO Identity Provider
- **Installing and Configuring O365 SSO Identity Provider**
  - Detailed explanations for installing the O365 SSO Identity Provider

What is the SurePassID O365 SSO Identity Provider?

The SurePassID O365 SSO Identity Provider (O365 IdP) is an SAML2 Identity Provider designed to provide advanced user authentication capabilities (MFA/Passwordless) capabilities to all your O365 applications.

The system is distributed as a Windows set-up install that you run on an existing 64-bit Windows server in your DMZ.

The O365 IdP supports on-premises and hybrid, private and public cloud installations of SurePassID Authentication Server.

The system supports both traditional OATH OTP password authentication, push technologies, as well as new technologies such as wearables, biometrics, and native FIDO U2F authentication on iOS, Android, and other mobile operating systems.

The system uses industry standard SAML2 authentication technologies.

For a primer on how Microsoft implements SAML2 Federated Identity Providers such as SurePassID, please review this Microsoft article.

Azure AD Connect: Use a SAML 2.0 Identity Provider for Single Sign On - Azure | Microsoft Docs

IMPORTANT: If you are planning to use ADFS you can install the SurePassID ADFS plug-in to add advanced authentication to ADFS.

Prerequisites: Windows Application Server

Application Server
A minimum of one application server is required to install the SurePassID O365 SSO Identity Provider. The application server can be run on-prem in your datacenter or in the cloud.

Depending on the number of users, authentication request load and fault tolerant requirements you will probably require more than one. More on that later.

The application server must be in the DMZ so that Microsoft Office 365 can make requests to the app server to authenticate users.

Supported Windows Systems
SurePassID O365 SSO Identity Provider can be installed on the following Windows versions:

- Windows Server 2016 – All versions
- Windows Server 2019 – All versions
- Windows 10
- Windows 11

Internet Information Server (IIS)
IIS must be installed on your Windows system. IIS is installed via the Webserver Role. In addition to IIS, the .NET 4.7 Feature must be enabled. You can follow these general steps if you are unfamiliar with installing IIS.

- Navigate to Administrative Tools and click Server Manager.
- In Server Manager, in the Roles Summary section, click "Add Roles and Features" to start the Add Roles Wizard and then click "Next".
- In the "Select Installation type page", select "Role-based or feature-based-installation" and click "Next"
- Select the server you will install IIS on and click "Next"
- When prompted to "Select server roles" check the "Webserver (IIS)" box then a pop-up window will come up.
- In the pop-up window, just click on "Add Features" then hit "Next".
- After that click "Next" on the next three consecutive windows selecting ".NET 4.7 Features"
- On the “Confirm installation selections” page click on “Install” and to perform the installation and when finished just click “Close“.

Load Balancing/Reverse Proxy
For performance and fault tolerance the SurePassID system can be installed on multiple servers behind load balancers/reverse proxies such as NGINX and F5 products for on-prem installs. If you are installing the app server in the cloud you can also make use of all Azure virtual network capabilities for application

isolation, Geographical (cross datacenter) load balancing systems like Azure Front Door/Azure Traffic and Azure Gateway for local load balancing and WAF capabilities.

Active Directory Connection
The app server will authenticate users first factor (username and password) in Active Directory and needs to be able to make local AD authentication requests.

Supported Windows Versions
Your O365 installation that must be configured as Hybrid Azure Active Directory and Password Hash Synchronization (PHS) and Pass-Through authentication (PTA) enabled. ADFS support is not required. For more info please visit:

https://docs.microsoft.com/en-us/azure/active-directory/hybrid/whatis-hybrid-identity

Prerequisites: Office 365

Office 365 Tenant – You must have an Office 365 tenant with a custom domain. It is advisable to create a non-production custom domain in your tenant to allow for testing MFA features without impacting the existing production users.

Global Admin – To perform the O365 installation (run PowerShell scripts against your O365 tenant) the user installing the system must have global admin rights. This should be in a non-federated domain such as your onmicrosoft.com domain. You will need this to rollback your changes to the custom domain that you federate later.

Security

SurePassID O365 IdP does not install with any IIS certificates for secure TLS operations but does require TLS 1.2 on the server. It is recommended that you configure the SurePassID O365 SSO Identity Provider (IIS web app) for SSL using corporate certificates for production. Alternatively, you may create self-signed certificates for testing.

Firewall Changes

The application server will communicate to any SurePassID MFA server to perform advanced authentication. For commercial accounts using SurePassID public cloud offering this would be the https://cloud2.suepassid.com endpoint. If you are using your own SurePassID MFA installation, then you would use your custom SurePassID MFA endpoint location.

In either case, the application server will require outbound firewall rules to allow TLS 1.2 or TLS 1.3 port connections to the SurePassID MFA server.

The SurePassID system is configured to communicate to the SurePassID MFA Authentication Server using transport level security (https).

Limit App Server Access

The system needs to be accessed by Microsoft O365 applications for the authentication of users.

Your firewall should be configured to only allow communications to SurePassID O365 SSO Identity Provider server from Microsoft IPs/domains.

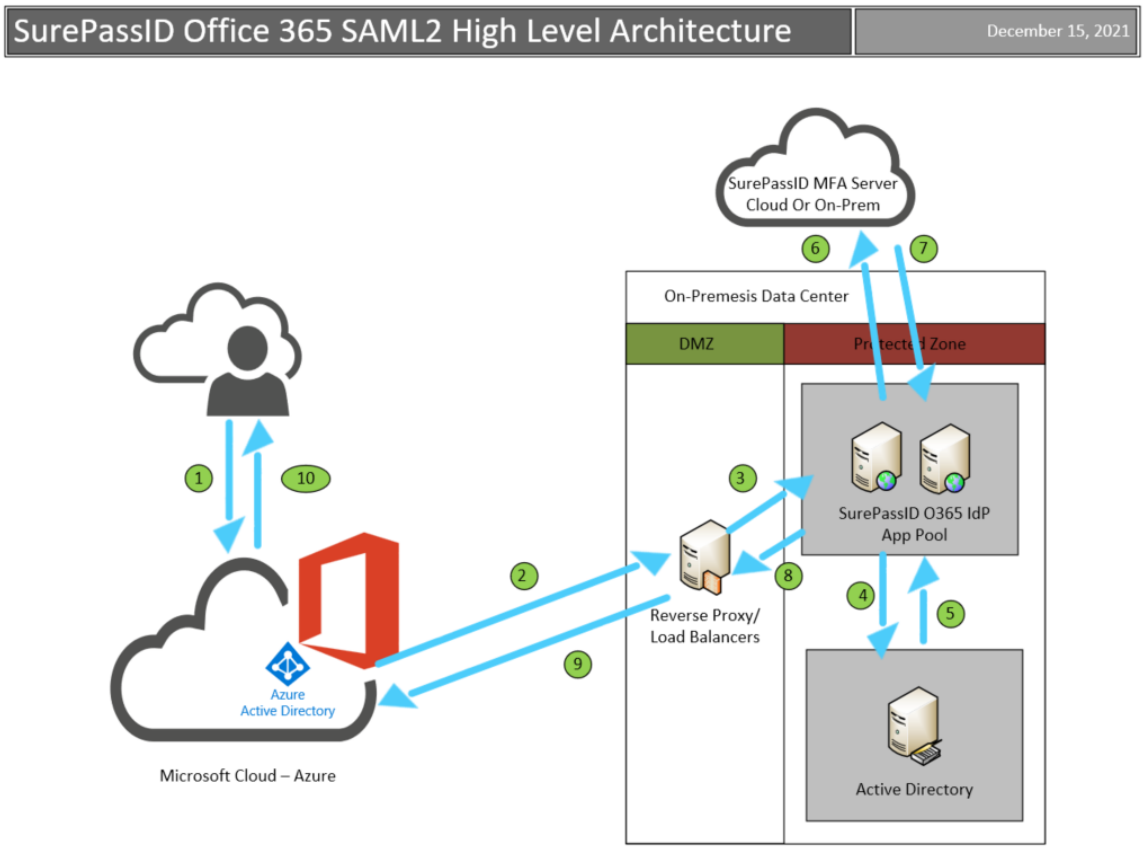
It is recommended that you follow security best practices for deploying and IIS web-based applications.

Request Logging

The system does support request logging. Logging captures the payload and IP of the requesting application. By default, the logs will be persisted in the local file system. Alternatively, they can be sent to the Windows Event Log for persistence, trouble shooting, or further analysis.

High-Level Architecture and Data Flow

The following diagram illustrates the high-level architecture of the system and authentication flow:



- The user logs into their office tenant app such as Teams, SharePoint, Outlook, etc.
- Azure AD redirects the user to SurePassID external endpoint (reverse proxy/load balancer) defined in the Azure AD Federated configuration file delivering a SAML2 Request.
- The external endpoint forwards the request to the SurePassID O365 IdP App pool.
- The SurePassID O365 IdP validates the users name and password with Active Directory.
- The SurePassID O365 IdP then validates the users second factor.
- After successful authentication SurePassID O365 IdP creates a SAML2 response, digitally signs it and sends the SAML2 response to Azure AD.
- Azure AD signs in the user and launches the Office 365 app that was originally requested by the user.

Installing and Configuring O365 SSO Identity Provider

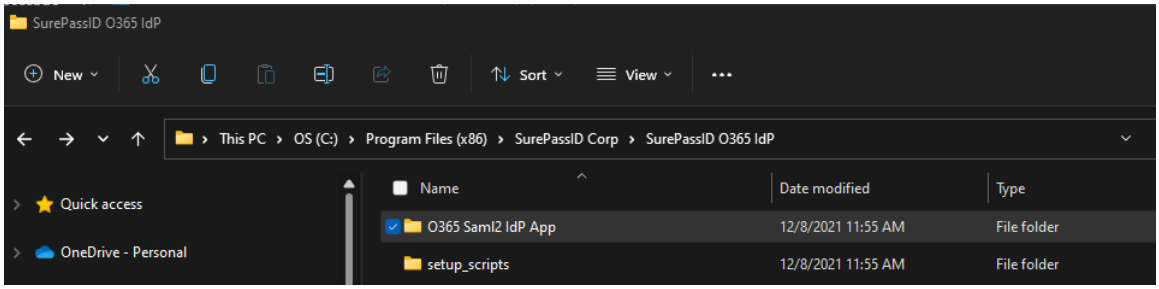
The SurePassID O365 SSO Identity Provider is distributed as a Windows exe installer (**SurePassIDO365Idp.exe**) located in a zip file (**SPO365Idp.ZIP**).

After downloading and unzipping **SPO365Idp.ZIP**, locate the file **SurePassIDO365Idp.exe** file, copy the file to the appropriate Windows server (if not already there) and run **SurePassIDO365Idp.exe** to install the system.

By default, the system installs into:

C:\Program Files (x86)\SurePassID Corp\SurePassID O365 IdP\ folder.

This folder contains two sub-folders:

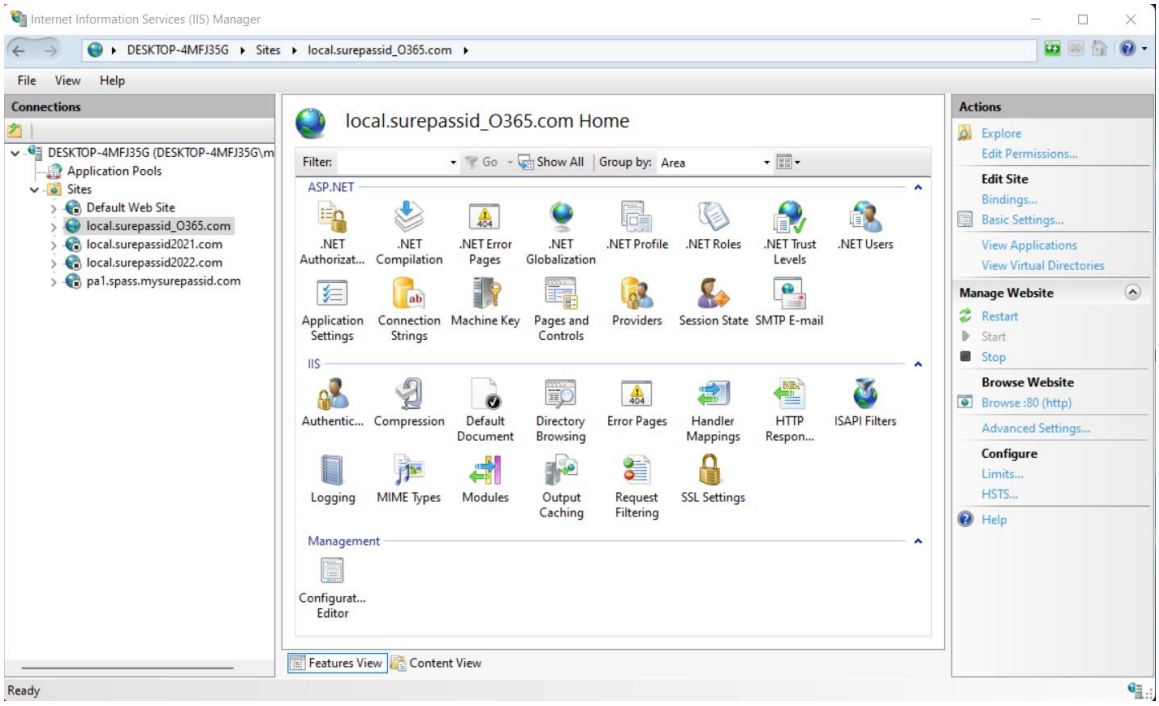


- O365 Saml2 IdP App folder – This is the SP O365 IdP web application that is installed as part of IIS. In subsequent steps you will modify the Web.config located in this folder.
- setup_scripts folder – This folder contains the following two PowerShell scripts.
  - o365_create_cert.ps1 – This PowerShell script creates the necessary files (such as certificates) to create a trust relationship with your O365 tenant.
  - install_Idp.ps1 – This PowerShell script sets up your O365 tenant to be SAML2 enabled

After the system is installed, you can proceed with the final installation steps that follow.

Complete IIS configuration

The installer will create a web site for the SurePassID O365 IdP App named local.surepassid_O365.com as shown below:



For the site to be operational you will need to do the following in IIS:

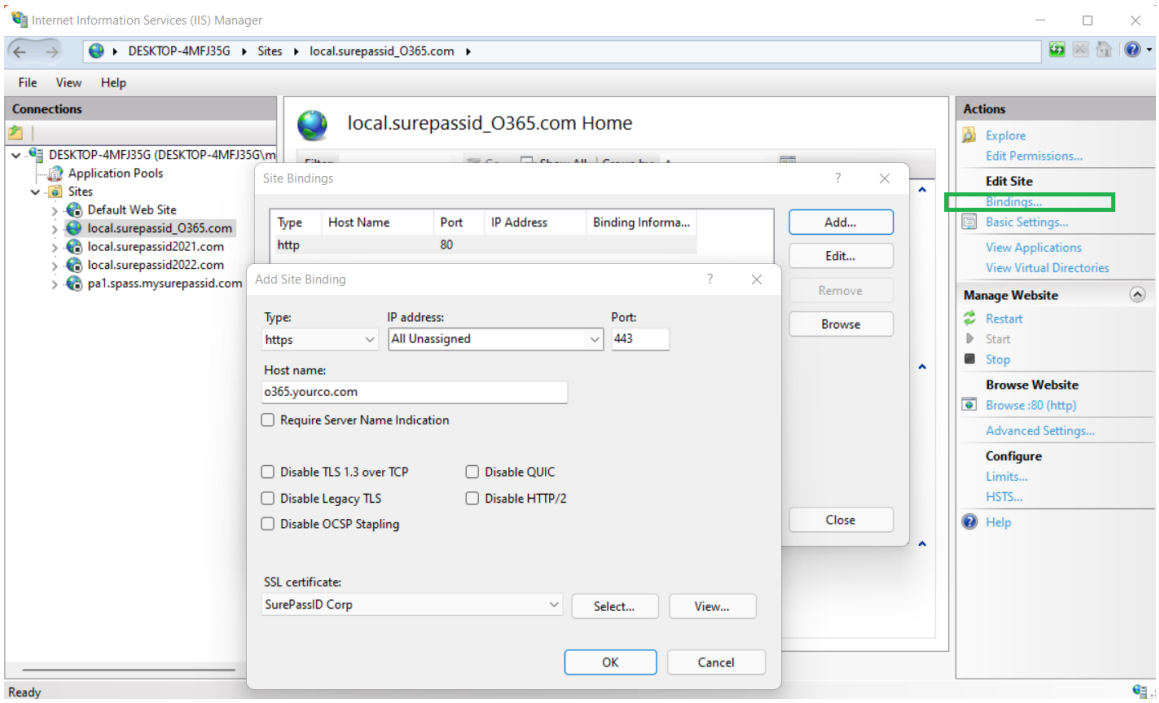
- Create an external DNS A record for site and point that to the IP of this server. For example, let select the external DNS name as o365.yourco.com.

- Procure an SSL certificate for the site o365.yourco.com using an approved CA. Do not use a self-signed or local CA certificate.

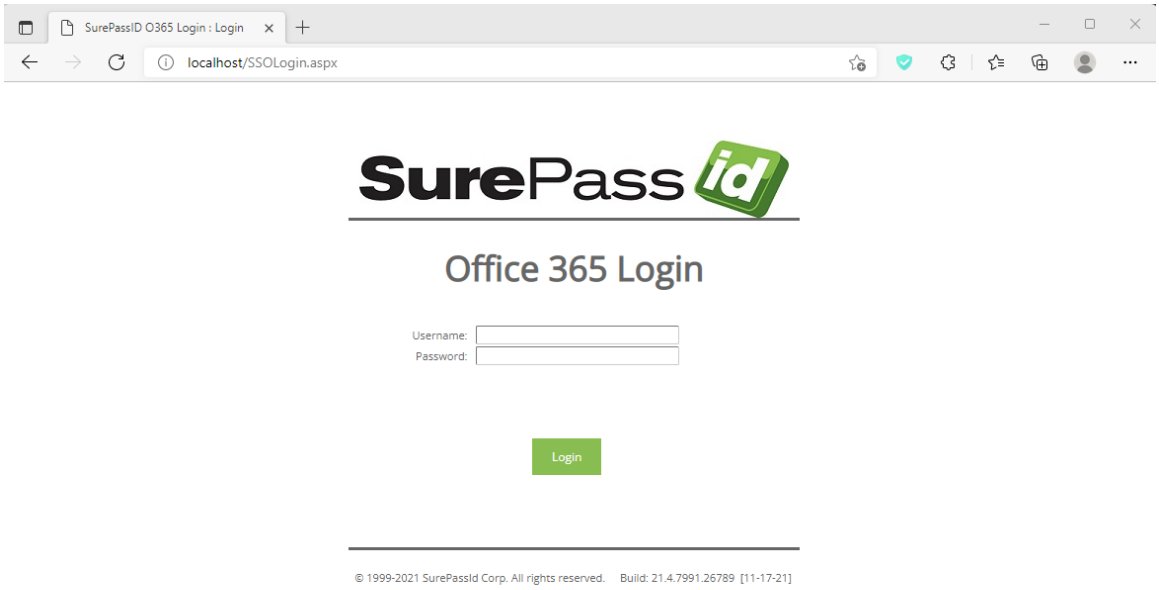
- Import that certificate into IIS.

  Use IIS to bind port 443 of o365.yourco.com using host headers. In IIS, select Bindings in the right most pane, click the Add button and set the following parameters.

  - The host header is the site's DNS name without https, such as o365.yourco.com.
  - Set the SSL certificate to the certificate you imported.



If the app is installed correctly, you will enter https://o365.yourco.com and see:



Then you are ready to proceed to the next step.

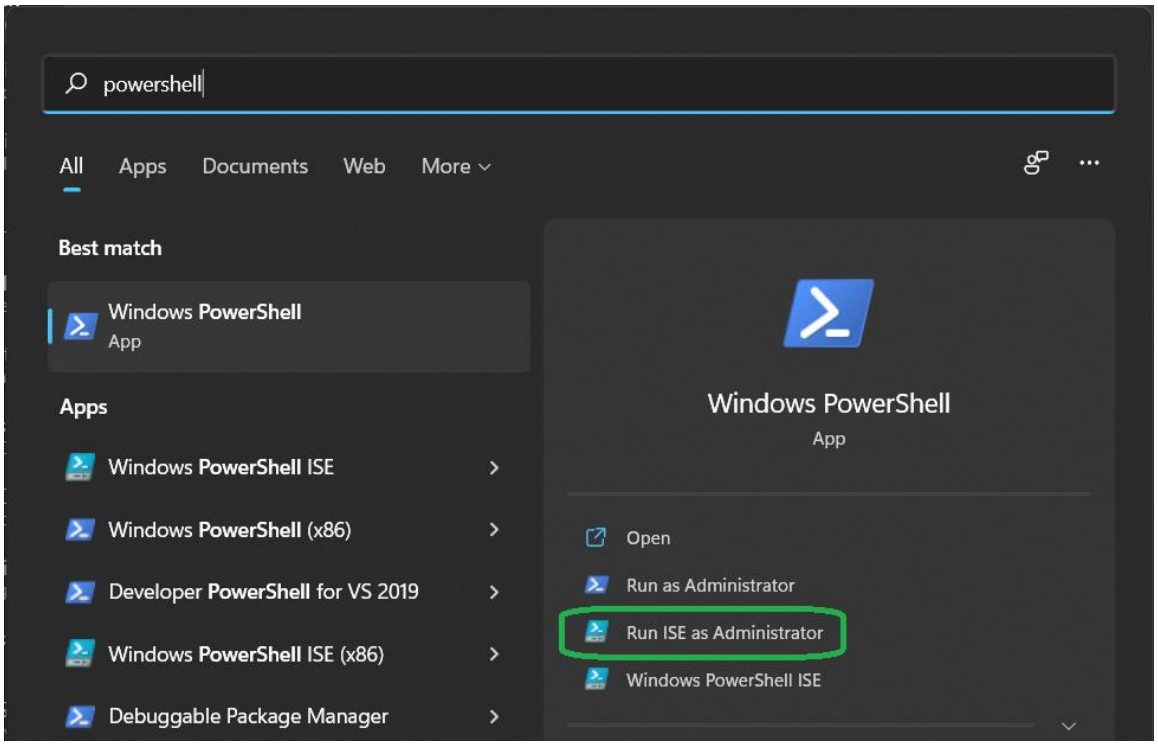
Generate O365 certs

To generate the O365 certs you will need to run the PowerShell script o365_create_cert.ps1 located in the setup-scripts folder in an elevated prompt. In

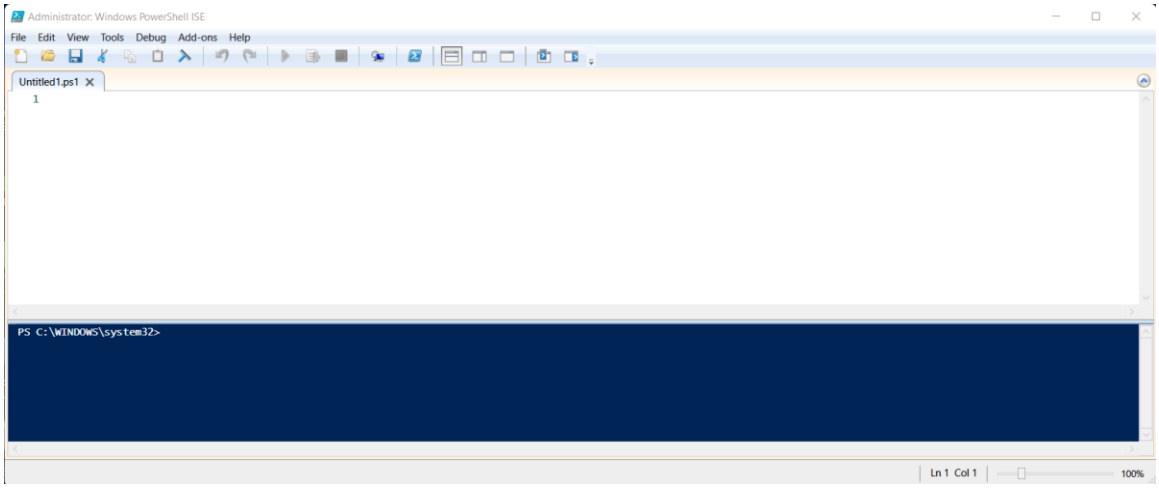
this example we use PowerShell ISE for illustrative purposes but there are many ways to run a PowerShell script, so use the method easiest for you. To start PowerShell ISE with elevated prompts, use these steps:

To open an elevated PowerShell prompt, in the taskbar search, type *powershell*.

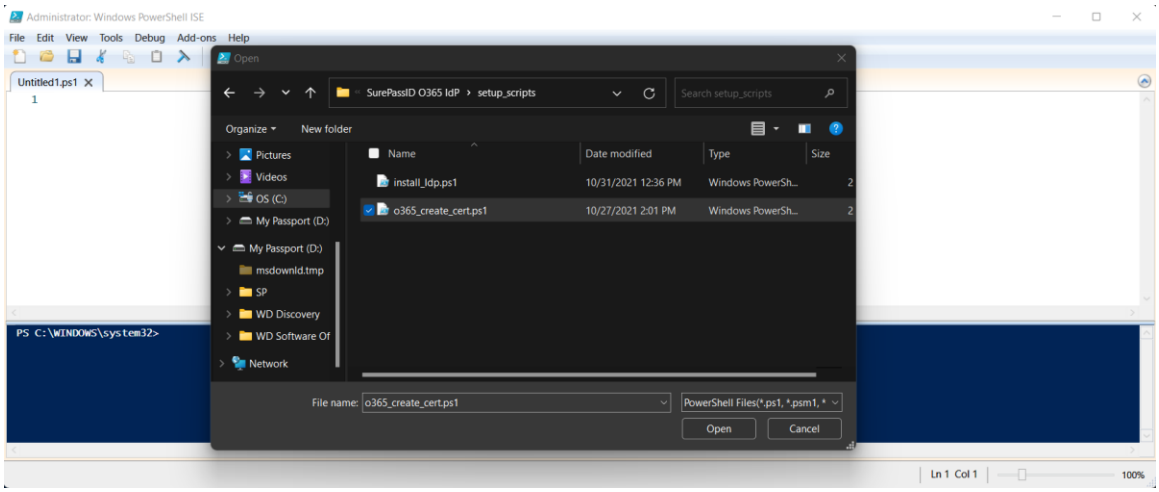
Now see the result *Windows PowerShell* which appears on the top. Right-click on it and select *Run ISA as Administrator* as shown below.



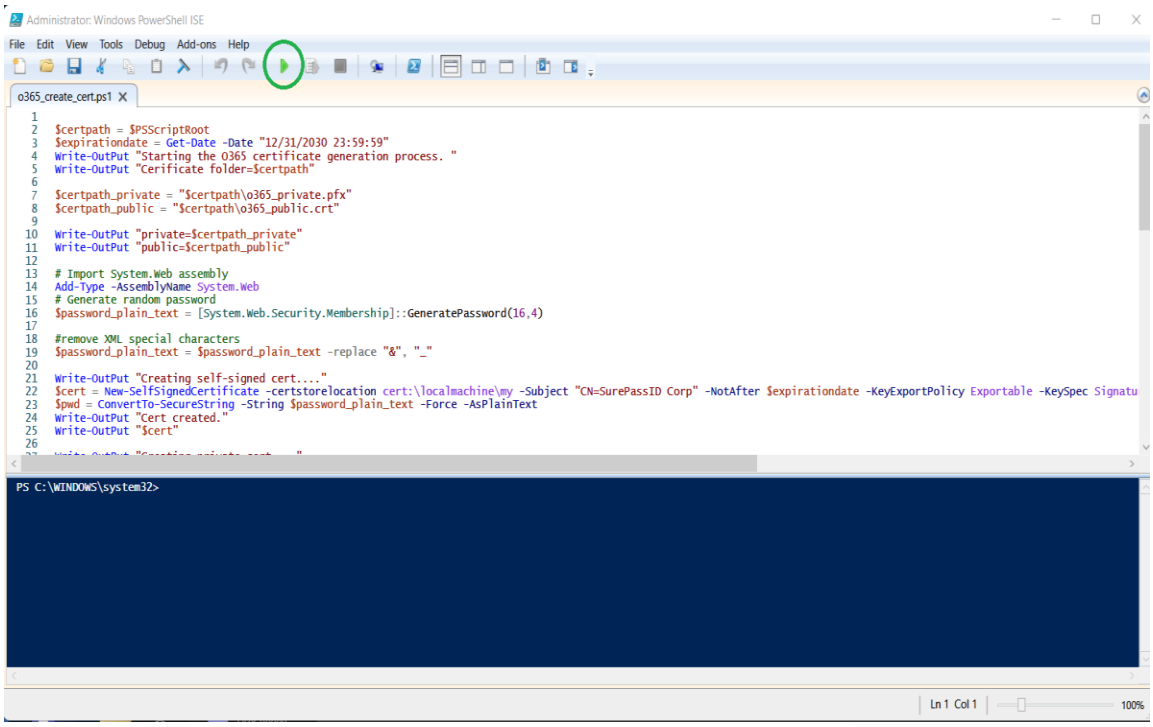
The following windows will be opened:



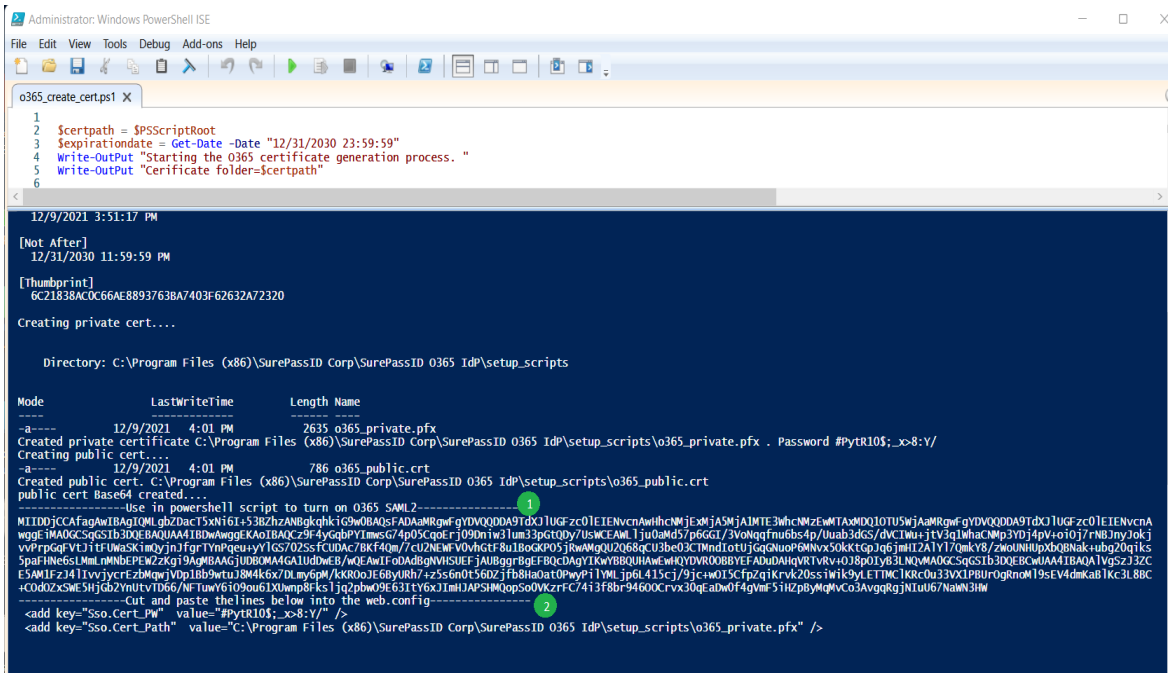
Select **File->Open** and navigate to the C:\Program Files (x86)\SurePassID Corp\SurePassID O365 IdP\setup_scripts folder and select the o365_create_cert.ps1



Click **Open** and the following window will be displayed. Press the green run button as highlighted below:



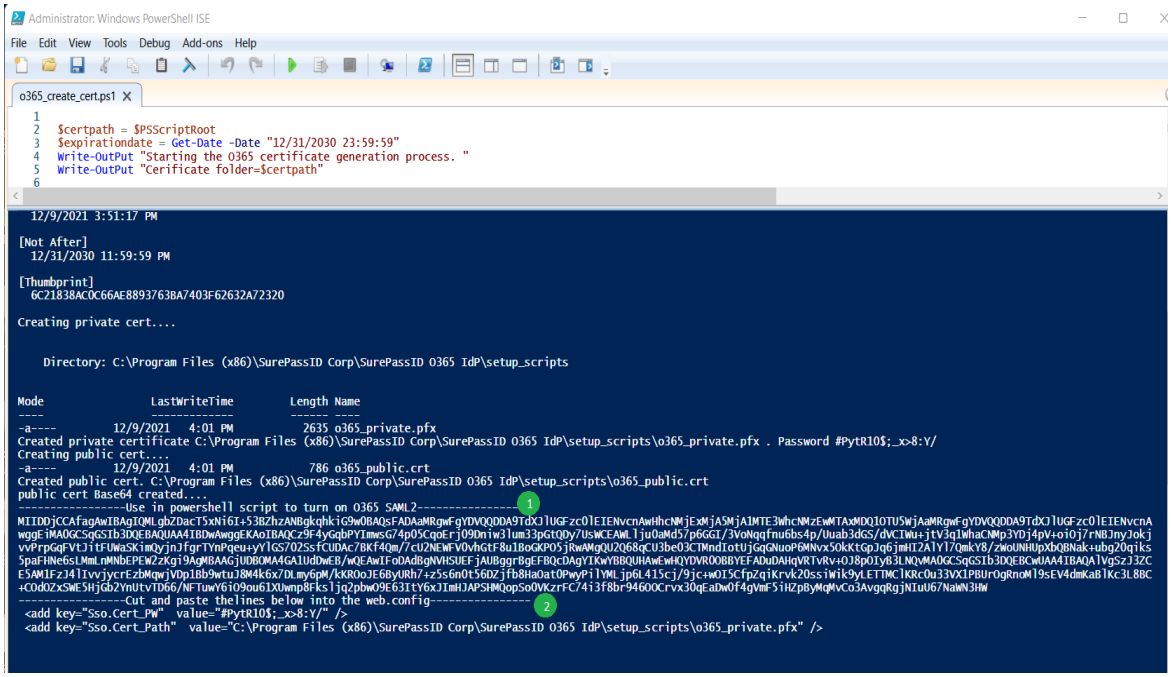
The script will start will run and display the following info:



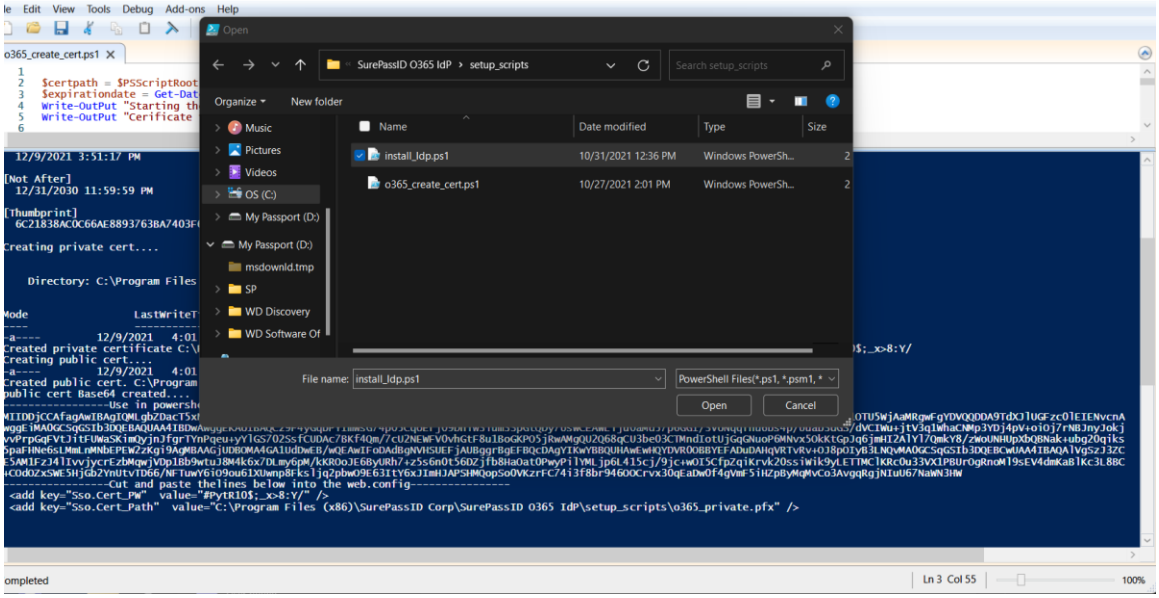
The data in section 1 will be used to update the certificate in the PowerShell script install_Idp.ps1 in the next step.

Section 2 will be used later to update web.config in the next step.
Do not close this window yet.

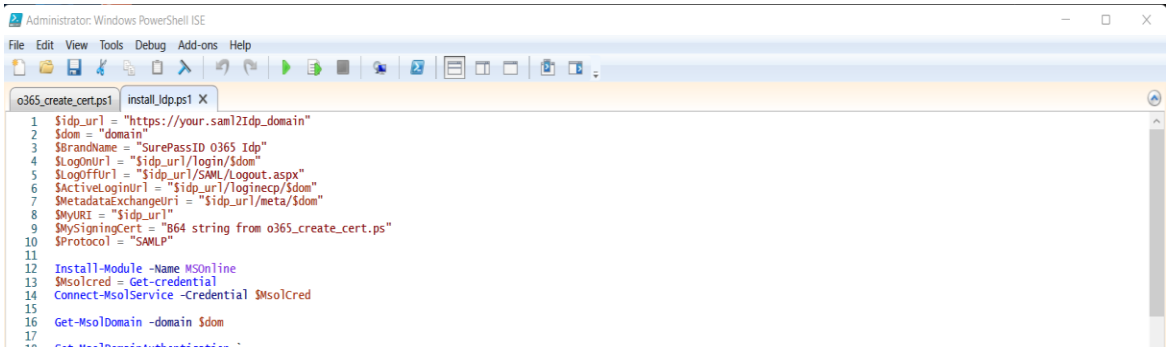
Copy the data in section 1 as shown below: (edit the screenshot below to put the second marker at the end of the crt data, up one line)



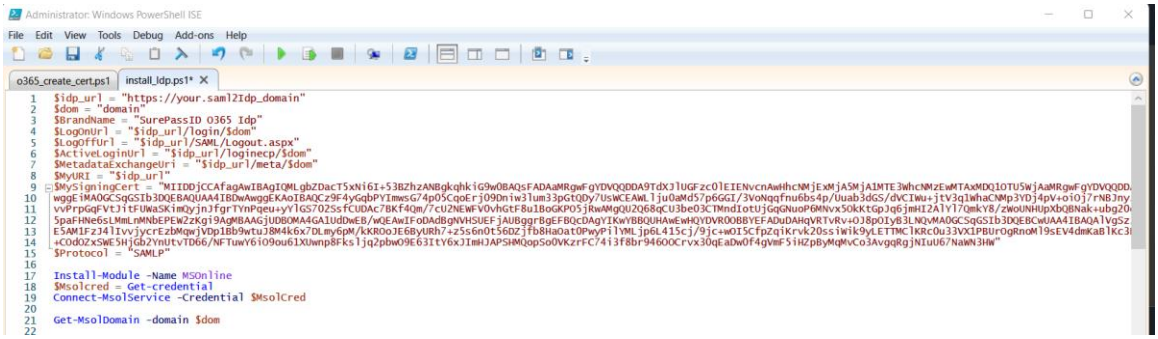
Open a new tab by select **File->Open** and navigate to the C:\Program Files (x86)\SurePassID Corp\SurePassID O365 IdP\setup_scripts folder and select the install_Idp.ps1 and click the **Open** button.



After opening install_Idp.ps1 scroll down to line 9 starting with $MySigningCert = “B64…..”



Paste the info copied from section 1 replacing the “**B64 string from o365_create_cert.ps”** text with info copied from section 1. Make sure to keep the beginning and ending double quotes. As shown below:



Next update line 1 by changing

$idp_url = “https://your.saml2idp.domain”

to the url you will using for this app for example:

$idp_url = https://o365.yourco.com

Next update line 2 by changing

$dom = “domain”

to the O365 custom domain you will be locking down with SurePassID. For example:

$dom = “abccorp.com”

Click **File->Save** to save install_idp.ps1

Very Important: This script will turn on SAML2 for your O365 tenant. This is not the time to run the script.

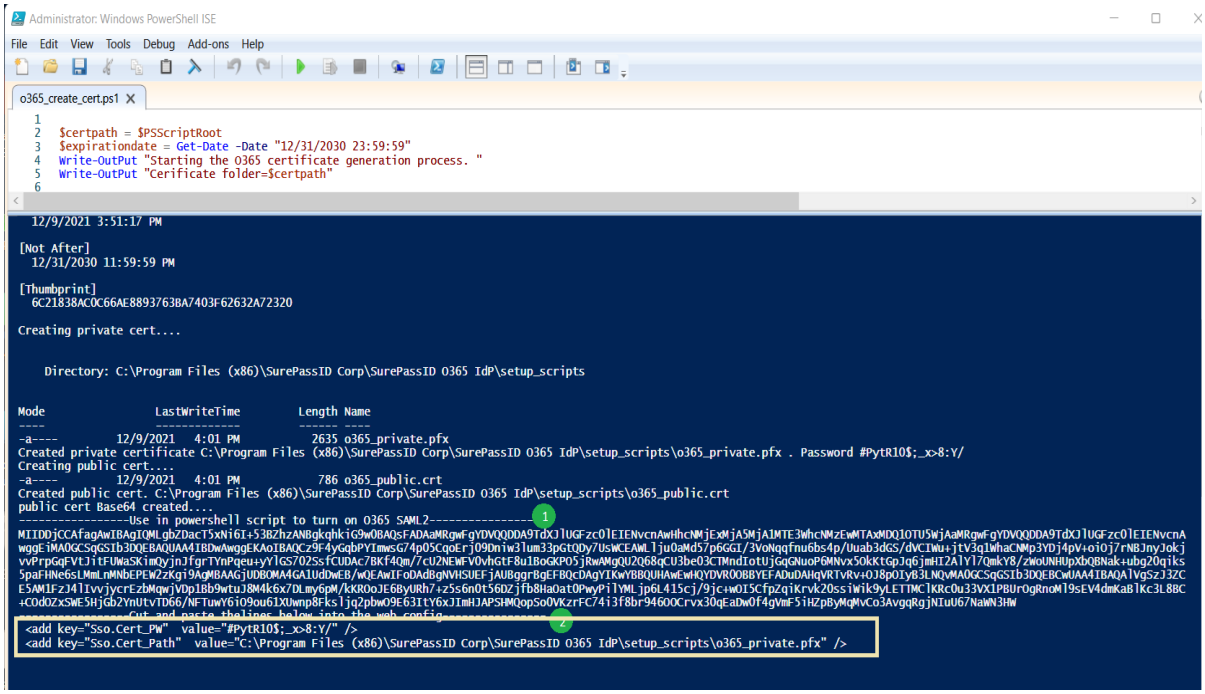
Configure SurePassID O365 App

The web.config file is an XML file and is part of the .net Framework. The file contains global customization settings. Some of the settings are SurePassID specific (<configuration><**appsettings>)** and you should change them to suit your needs. Other settings effect the way that ASP .net operates and you should not change these settings unless you have experience in this area. Some settings you can change and others you should not. If you make a change to web.config that violates the rules of xml syntax, the system will not run and you will receive an IIS error.

Let's start by opening the **web.config** file located in C:\Program Files (x86)\SurePassID Corp\SurePassID O365 IdP\O365 Saml2 IdP App folder.

```
<?xml version="1.0"?>
<configuration>
  <appSettings>

    <!-- SurepassID MFA server endpoint and your tenant sandbox by default -->
       <add key="Server.RESTEndPoint" value="https://sandbox.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/>
    <!-- SurepassID MFA server production endpoint -->
       <!--add key="Server.RESTEndPoint" value="https://cloud2.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/-->
       <add key="Server.CompanyAccount" value=""/>
       <add key="Server.CompanyAccountKey" value=""/>

    <!-- use the confguration tool to create the private key certificate and password -->
       <add key="Sso.Cert_PW" value=""/>
       <add key="Sso.Cert_Path" value="C:\Program Files (x86)\SurePassID\setup_scripts\o365_private.pfx"/>

    <!-- saml2 settings -->
    <add key="Sso.SurePassID_Issuer" value="idp url"/>
    <add key="Sso.LoginGoodFor_HH:MM:SS" value="1:00:00"/>
    <add key="Sso.RP_Issuer" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.Audience" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.ACS" value="https://login.microsoftonline.com/login.srf"/>

     <!-- directory settings  -->
    <add key="Server.DefaultDomain" value="your default domain" />
    <add key="Server.DirectoryEndpointType" value="AD"/>

     <!-- push settings  -->
     <add key="PushAccountName" value="SurePassID O365 IdP"/>

     <!-- fido settings   -->
    <add key="Server.AppId" value="https://cloudsso.surepassid.com"/>

     <!-- debug settings   -->
    <add key="Server.Trace" value="0"/>
    <add key="Server.TracePath" value=""/>
```

Copy the data from section 2 replacing the data in the green section below in the green section in web.config a shown below: (the screenshot below should be larger as it is hard to see the section clearly)




```
<?xml version="1.0"?>
<configuration>
  <appSettings>

    <!-- SurepassID MFA server endpoint and your tenant sandbox by default -->
       <add key="Server.RESTEndPoint" value="https://sandbox.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/>
    <!-- SurepassID MFA server production endpoint -->
       <!--add key="Server.RESTEndPoint" value="https://cloud2.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/-->
       <add key="Server.CompanyAccount" value=""/>
       <add key="Server.CompanyAccountKey" value=""/>

    <!-- use the configuration tool to create the private key certificate and password -->
       <add key="Sso.Cert_PW" value=""/>
       <add key="Sso.Cert_Path" value="C:\Program Files (x86)\SurePassID\setup_scripts\o365_private.pfx"/>

    <!-- saml2 settings -->
    <add key="Sso.SurePassID_Issuer" value="idp url"/>
    <add key="Sso.LoginGoodFor_HH:MM:SS" value="1:00:00"/>
    <add key="Sso.RP_Issuer" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.Audience" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.ACS" value="https://login.microsoftonline.com/login.srf"/>

     <!-- directory settings  -->
    <add key="Server.DefaultDomain" value="your default domain" />
    <add key="Server.DirectoryEndpointType" value="AD"/>

     <!-- push settings  -->
     <add key="PushAccountName" value="SurePassID O365 IdP"/>

     <!-- fido settings   -->
    <add key="Server.AppId" value="https://cloudsso.surepassid.com"/>

     <!-- debug settings   -->
    <add key="Server.Trace" value="0"/>
    <add key="Server.TracePath" value=""/>
```

On line 17 in web.config set the **Sso.SurePassID_Issuer** parameter to be in the format of:

https://$idp_url/meta/$dom

where $idp_url and $dom are set in the preceding step .

For example:

The Issuer value should look like: <add key="Sso.SurePassID_Issuer" value="https://o365.yourco.com/meta/ abccorp.com "/>

```
<?xml version="1.0"?>
<configuration>
  <appSettings>

    <!-- SurepassID MFA server endpoint and your tenant sandbox by default -->
       <add key="Server.RESTEndPoint" value="https://sandbox.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/>
    <!-- SurepassID MFA server production endpoint -->
       <!--add key="Server.RESTEndPoint" value="https://cloud2.surepassid.com/AuthServer/REST/U2F/U2FServer.aspx"/-->
       <add key="Server.CompanyAccount" value=""/>
       <add key="Server.CompanyAccountKey" value=""/>

    <!-- use the confguration tool to create the private key certificate and password -->
       <add key="Sso.Cert_PW" value=""/>
       <add key="Sso.Cert_Path" value="C:\Program Files (x86)\SurePassID\setup_scripts\o365_private.pfx"/>

    <!-- saml2 settings -->
    <add key="Sso.SurePassID_Issuer" value="https://o365.yourco.com/meta/yourO365domain"/>
    <add key="Sso.LoginGoodFor_HH:MM:SS" value="1:00:00"/>
    <add key="Sso.RP_Issuer" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.Audience" value="urn:federation:MicrosoftOnline"/>
    <add key="Sso.ACS" value="https://login.microsoftonline.com/login.srf"/>

     <!-- directory settings  -->
    <add key="Server.DefaultDomain" value="your default domain" />
    <add key="Server.DirectoryEndpointType" value="AD"/>

     <!-- push settings  -->
     <add key="PushAccountName" value="SurePassID O365 IdP"/>

     <!-- fido settings   -->
    <add key="Server.AppId" value="https://o365.yourco.com"/>

     <!-- debug settings   -->
    <add key="Server.Trace" value="0"/>
    <add key="Server.TracePath" value=""/>
```

You will need to update the remaining config parameters that follow:

The table below describes the SurePassID specific settings. The items highlighted in red you must set, items highlighted in green were set in preceding steps and items highlighted in tan are optional.

<configuration>
<appsettings> keys

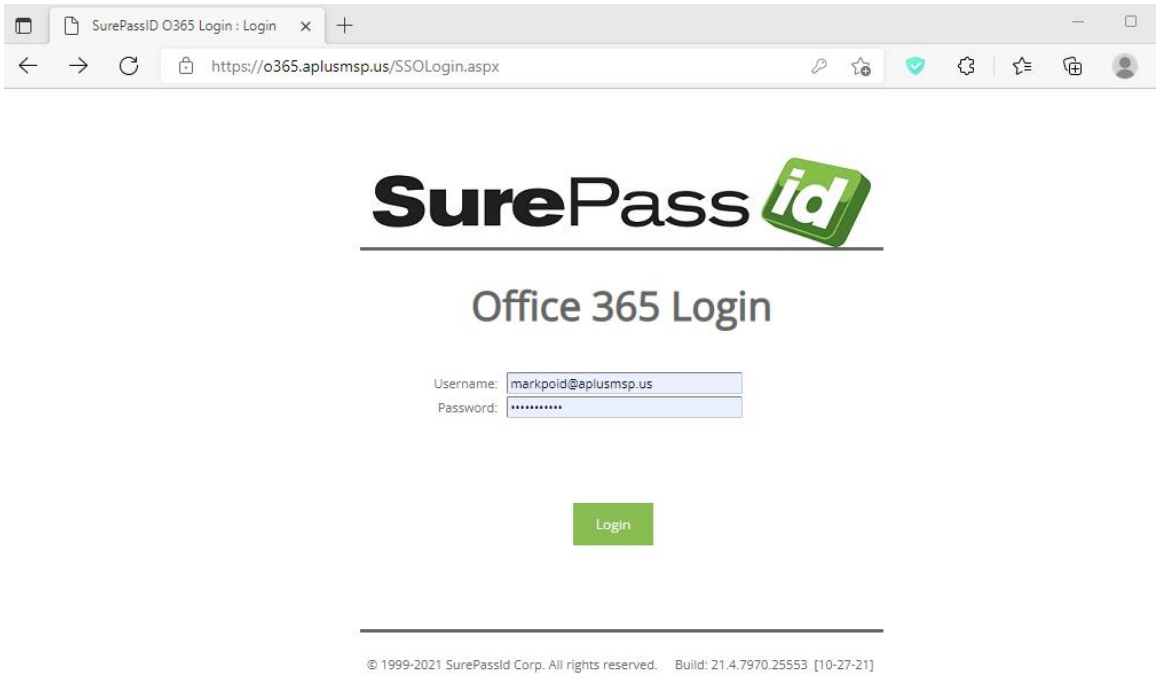
Parameter	Description
Server.RESTEndPoint	The SurePassID MFA server endpoint that will process all mobile requests.
Server.CompanyAccount	Your company account in the SurePassID MFA server
Server.CompanyAccountKey	Your company account key in the server
Sso.Cert_PW	Updated in previous step. Do not change.
Sso.Cert_Path	Updated in previous step. Do not change.
Sso.SurePassID_Issuer	Updated in previous step. Do not change.
Sso.LoginGoodFor_HH:MM:SS	SAML2 assertion lifetime. Currently ignored by Azure AD.
Sso.RP_Issuer	Do not change.
Sso.Audience	SAML2 audience. Do not change.
Sso.ACS	SAML2 assertion consumer service. Do not change.
Server.DefaultDomain	Set to your local Active Directory domain
Server.DomainRoutingPath	Set this to the path to the domain routing table. Domain routing allows user to be authenticated different local Acitve Directory systems.
Server.DirectoryEndpointType	Set to AD. Do not change.
Server.AppId	Fido AppID. Set to the URL of the SAML2 app. For example: https://o365.yourco.com
Server.Trace	Log all activity that passes through the server. The output is provided in the **Trace** subfolder of the installation folder. 0=OFF 1-ON. Only use for debugging.

Server.TracePath	The location of the trace file.

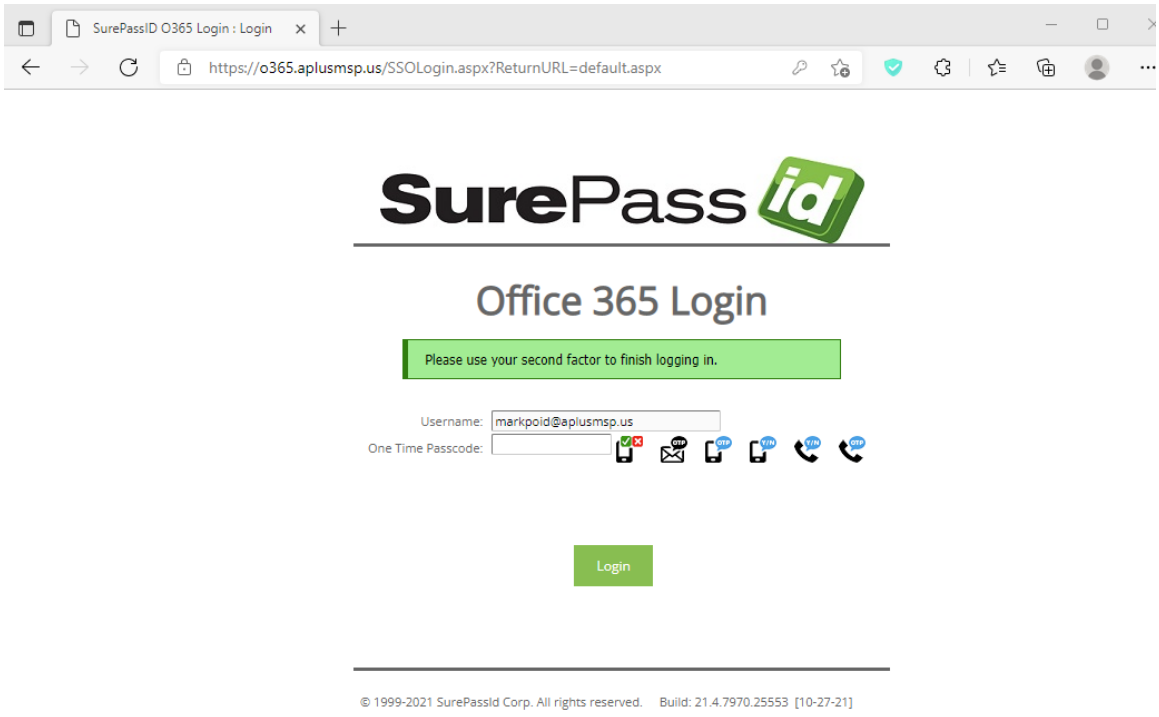
Installation Verification

You should now be able to verify that the system is set-up correctly and can connect to SurePassID MFA services. To verify the system is operating correctly, you will need to follow these steps:

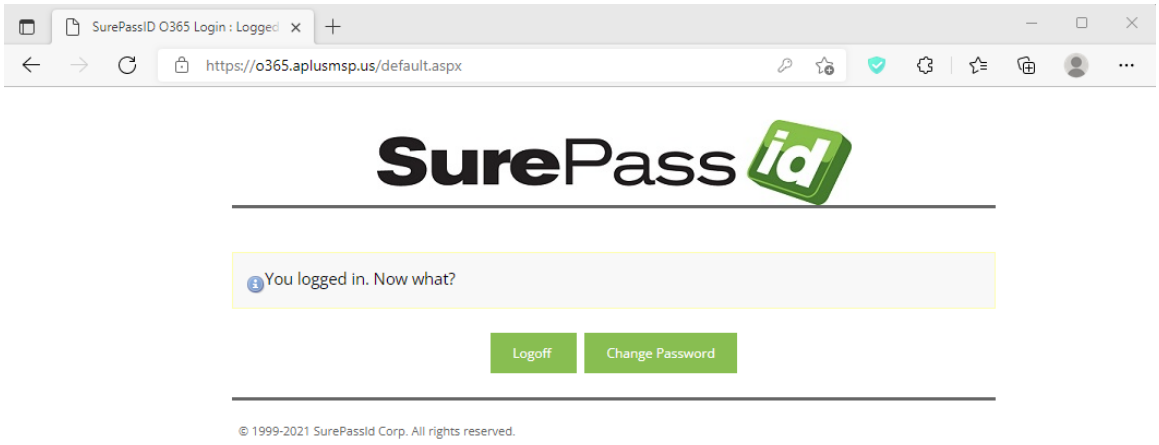
Start the app by entering the URL of the application such as https://o365.yourco.com and enter your AD credentials as show below.



Press **Login** button. If your credentials are correct, then you will be presented with the following form to use your second factor of authentication as shown below:



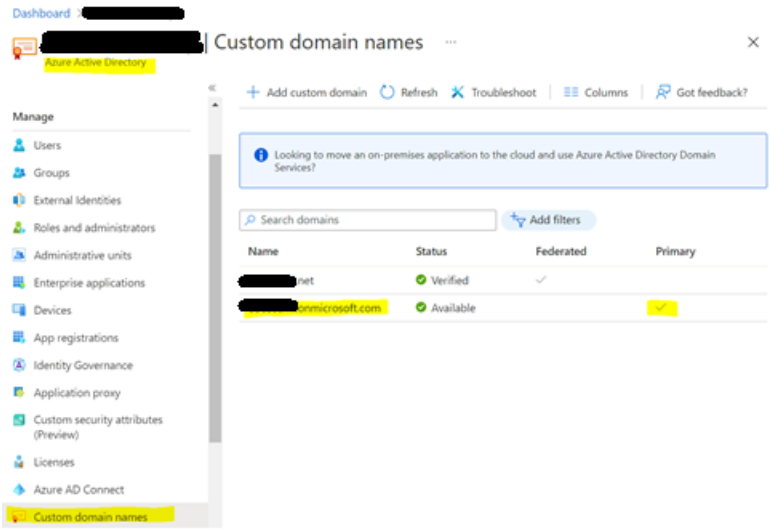
After authenticating with your second factor, you will see the following form:



Great job! You are ready to turn on MFA for your O365 tenant. You can click **Logoff** now.

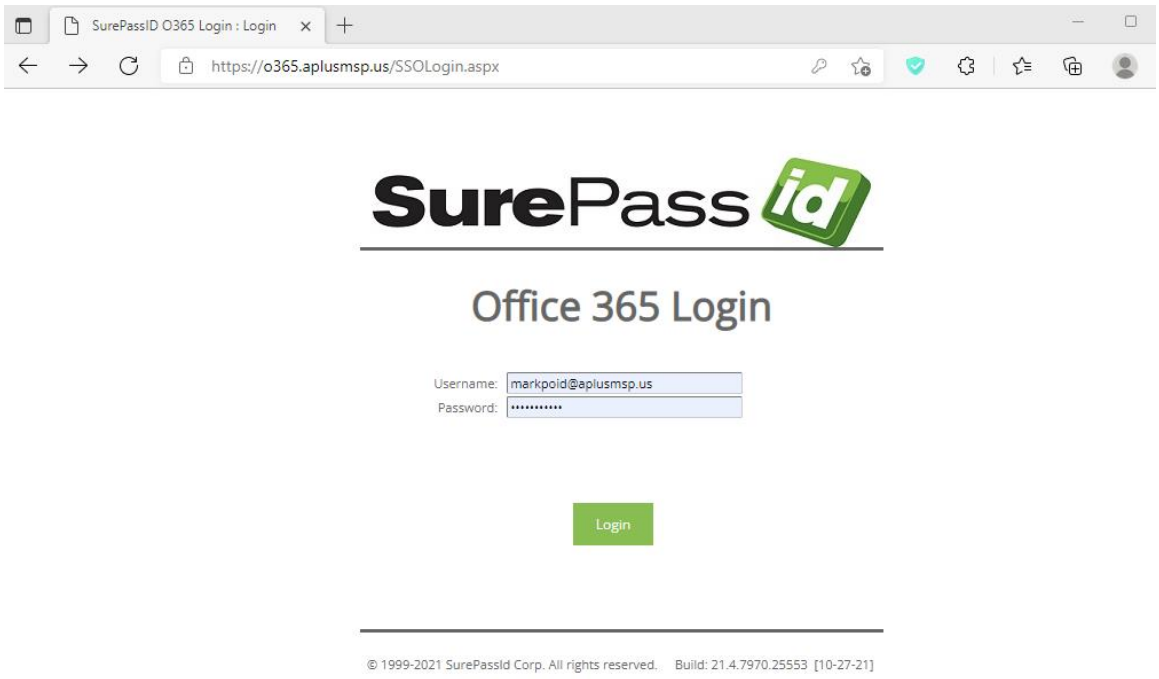
Update your O365 tenant to use SP O365 MFA

Note that before running the install_idp.ps1 script you may need to set another domain as the default domain in your O365 tenant as shown here:



To update your O365 tenant to allow SurePassID MFA you will need to run the PowerShell script install_Idp.ps1. You will need to be a Global Administrator to run the script. We use the .onmicrosoft.com domain global administrator account for this example.

After running the install_Idp.ps1 script when you login to any of your O365 apps, re-authentication will be required:



After you authenticate, you will then be redirected back to O365 and logged into the O365 app you originally requested.

If you run into problems logging in to your account, you can always reset your Azure AD back to Manual (not federated) and login like you have in the past by running the following PowerShell commands:

Set-MsolDomainAuthentication -DomainName $dom -Authentication Managed

The following command can verify that your O365 tenant is back to the way it was prior to installing SurePassID SAML2 federation.

Get-MsolDomainFederationSettings -DomainName $dom

These commands are commented out at the bottom of install_idp_install.ps1 script.

Domain Routing

For most systems all your O365 users will be authenticated.