How to create an Authentication Provider for Office365

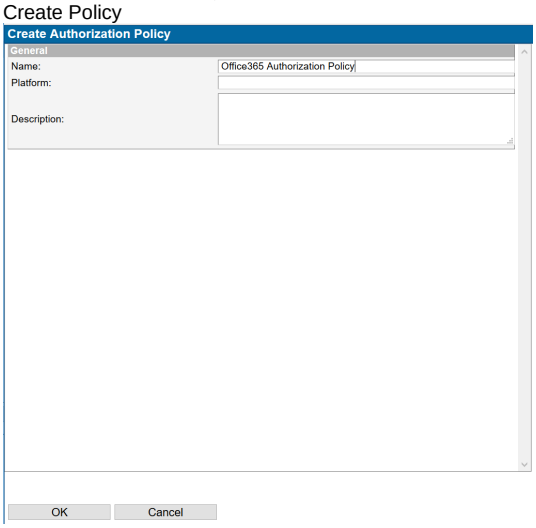
To allow users authenticated using Ubisecure SSO to log into Office365 applications, or any Azure AD connected application.

Step-by-step guide

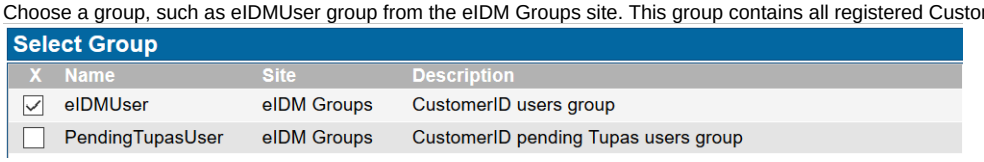
To configure an IDP for Office365, complete the following steps:

1. Create an Agent
    a. SAML Application
        - Use compatibility flags: `AuthnRequestValidate AssertionSignCertificate HttpPostResponseSign`

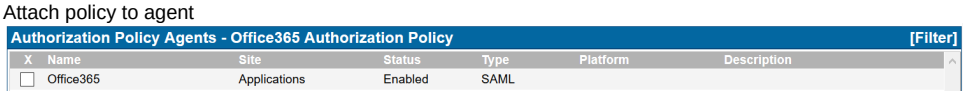
2. Create a new Authorization Policy
    a. Create Policy



    b. Press OK
    c. Select the Attributes tab, select Add...
    d. Choose a group, such as eIDMUser group from the eIDM Groups site. This group contains all registered CustomerID users.



    e. Add the following attributes
        - `${nameID.format('persistent').value('ImmutableID')}`
        - `${attribute.nameFormat(null).name('IDPEmail').values('Mail')}`
        - `${issuer.value('Issuer')}`
    f. Attach policy to agent



    g. Complete the rest of the appropriate settings for access control
3. Activate the agent in Azure AD
    a. Set Azure AD domain to Federated mode with Ubisecure SSO as IDP
        i. Use the model powershell script to activate using the Azure APIs

```
$upn = "admin@ubidemo2.onmicrosoft.com"
$file = "$($env:USERPROFILE)\AzureAD\$upn.txt"
$credendial = $null
if(Test-Path $file) {
    $credendial = [pscredential]::new($upn, (Get-Content -Path $file | ConvertTo-
SecureString))
} else {
    $dir = Split-Path -Parent -Path $file
    New-Item -Force -ItemType Directory -Path $dir | Out-Null
    $credendial = Get-Credential -Message $upn -UserName $upn
    $file = Join-Path -Path $dir -ChildPath "$($credendial.UserName).txt"
    ConvertFrom-SecureString -SecureString $credendial.Password | Set-Content -Path $file -
Force
}
```

```
Connect-MsolService -Credential $credendial

#$sso = "https://sso.ubidemo1.com"
$sso = "https://gmo.ubidemo.com"

$metadata = Invoke-WebRequest -Uri "$sso/uas/wsf/FederationMetadata.xml" -UseBasicParsing |
Select-Object -ExpandProperty Content
$cert = ([xml]$metadata).EntityDescriptor.RoleDescriptor.KeyDescriptor.KeyInfo.X509Data.
X509Certificate
$cert = [Convert]::ToBase64String([System.Security.Cryptography.X509Certificates.
X509Certificate]::new([Convert]::FromBase64String($cert)).GetRawCertData())

#$MetadataExchangeUri = "$sso/uas/wsf/FederationMetadata.xml"
$MetadataExchangeUri = "$sso/uas"
#$MetadataExchangeUri = "urn:null"
$IssuerUri = "https://ubidemo2.com"
#$IssuerUri = "$sso/uas"

if($false) {
    Remove-MsolUser -UserPrincipalName "koskmaar@ubidemo1.com" -Force:$true
    Remove-MsolUser -UserPrincipalName "koskmaar@ubidemo1.com" -Force:$true -
RemoveFromRecycleBin
    Remove-MsolDomain -DomainName "ubidemo2.com" -Force:$true

    New-MsolDomain -Name "ubidemo2.com" -VerificationMethod DnsRecord
    Get-MsolDomainVerificationDns -DomainName "ubidemo2.com"
    Confirm-MsolDomain -DomainName "ubidemo2.com"
}

if($false) {
    New-MsolUser -UserPrincipalName "koskmaar@ubidemo1.com" -DisplayName "Maarit Koskinen" -
ImmutableId "6HNhl3wbx0GSrkC96VWQ0g==" -FirstName "Maarit" -LastName "Koskinen"
}

if($false) {
    Set-MsolDomainAuthentication -Authentication Managed -DomainName "ubidemo2.com"
}

if($false) {
    Set-MsolDomainAuthentication `
        -Authentication Federated `
        -DomainName "ubidemo2.com" `
        -ActiveLogOnUri "$sso/uas/saml2/soap/SingleSignOnService" `
        -FederationBrandName "ubidemo2.com" `
        -IssuerUri $IssuerUri `
        -LogOffUri "$sso/uas/logout" `
        -MetadataExchangeUri $MetadataExchangeUri `
        -NextSigningCertificate $null `
        -OpenIdConnectDiscoveryEndpoint $null `
        -PassiveLogOnUri "$sso/uas/saml2/SingleSignOnService" `
        -PreferredAuthenticationProtocol Samlp `
        -SigningCertificate $cert `
        -Verbose

    Set-MsolDomainFederationSettings `
        -DomainName "ubidemo2.com" `
        -ActiveLogOnUri "$sso/uas/saml2/soap/SingleSignOnService" `
        -FederationBrandName "ubidemo2.com" `
        -IssuerUri $IssuerUri `
        -LogOffUri "$sso/uas/logout" `
        -MetadataExchangeUri $MetadataExchangeUri `
        -NextSigningCertificate $null `
        -OpenIdConnectDiscoveryEndpoint $null `
        -PassiveLogOnUri "$sso/uas/saml2/SingleSignOnService" `
        -PreferredAuthenticationProtocol Samlp `
        -SigningCertificate $cert `
        -Verbose
}

Set-MsolDomainFederationSettings `
```

```
    -DomainName "ubidemo2.com" `
    -ActiveLogOnUri "$sso/uas/saml2/soap/SingleSignOnService" `
    -FederationBrandName "ubidemo2.com" `
    -IssuerUri $IssuerUri `
    -LogOffUri "$sso/uas/logout" `
    -NextSigningCertificate $null `
    -PassiveLogOnUri "$sso/uas/saml2/SingleSignOnService" `
    -PreferredAuthenticationProtocol Samlp `
    -SigningCertificate $cert `
    -Verbose

Get-MsolDomainFederationSettings -DomainName "ubidemo2.com" | fl
```

b. remove certificates from metadata when activating

This enables sign-in using Modern Authentication, supported by modern Office fat client applications. Legacy clients using WS-Federation Active Profile are not supported.

Related articles

- How to create an Authentication Provider for Office365