MS Teams Online Meetings Integration with 3rd-Party Apps (MS Graph API)

Overview

Overview

This configuration enables a third-party application to create and manage Microsoft Teams online meetings using Microsoft Graph API integration.

The integration is typically used by:

- Student engagement platforms
- Recruitment systems
- CRM platforms
- Booking systems
- Service management applications
- Virtual appointment platforms

The implementation uses:

- Microsoft Entra ID App Registration
- Microsoft Graph API
- Teams OnlineMeetings API
- Teams Application Access Policies

This approach allows controlled programmatic creation of Teams meeting links while maintaining governance and restricting which user accounts the application may act on.

This configuration is important because it:

- Enables secure automation of Teams meeting creation
- Prevents unrestricted tenant-wide meeting creation by third-party apps
- Provides governance and operational control
- Reduces the blast radius in the event of application compromise

> Prerequisites

Licensing

Ensure the following licenses/services are available:

- Microsoft 365 tenant
- Microsoft Teams enabled
- Microsoft Entra ID
- Microsoft Graph API access

Required Roles

The implementing administrator should have:

- Global Administrator OR
- Application Administrator
- Teams Administrator

Required PowerShell Modules

Install the following PowerShell modules:

```
Install-Module Microsoft.Graph -Scope CurrentUser
Install-Module MicrosoftTeams -Scope CurrentUser
```

Required Permissions

The Entra ID application registration will require:

Microsoft Graph Application Permissions

- `OnlineMeetings.ReadWrite.All`
- `User.Read.All` OR `User.ReadBasic.All`

Admin consent must be granted.

Certificate Authentication (Recommended)

Certificate-based authentication is strongly recommended over client secrets for:

- Improved security
- Reduced credential exposure
- Better long-term automation support
- Reduced secret rotation overhead

Step 1: Create the Entra ID Application Registration

Navigate to:

Entra Admin Center > Applications > App registrations

Create the Application

Select:

- New registration

Configure:

- Name: `Teams Meetings Integration`
- Supported account type: Single tenant (recommended)

Select:

- Register

Record the following values

Save:

- Application (client) ID
- Directory (tenant) ID

These values will be required for:

- Graph authentication
- PowerShell automation
- Third-party application configuration

Step 2: Configure API Permissions

Navigate to:

App Registration > API permissions

Add Microsoft Graph Application Permissions

Add:

- `OnlineMeetings.ReadWrite.All`
- `User.Read.All`

OR for reduced exposure:

- `User.ReadBasic.All`

Grant Admin Consent

Select:

- Grant admin consent for tenant

Validate Permission Status

Ensure all permissions display:

- `Granted for <TenantName>`

Step 3: Configure Certificate Authentication

Generate Certificate

Run PowerShell:

```
$cert = New-SelfSignedCertificate `
    -Subject "CN=TeamsMeetingsIntegration" `
    -CertStoreLocation "Cert:\CurrentUser\My" `
    -KeySpec Signature `
    -KeyLength 2048 `
    -KeyExportPolicy Exportable `
    -HashAlgorithm SHA256 `
    -NotAfter (Get-Date).AddYears(2)
```

Export Public Certificate

```
Export-Certificate `
    -Cert $cert `
    -FilePath "C:\Temp\TeamsMeetingsIntegration.cer"
```

Upload Certificate to App Registration

Navigate to:
App Registration > Certificates & secrets > Certificates

Upload:

- `.cer` file

Record Certificate Thumbprint

Run:

```
$cert.Thumbprint
```

Save the thumbprint securely.

Step 4: Create the Teams Application Access Policy

Connect to Microsoft Teams PowerShell

```
Connect-MicrosoftTeams
```

Create the Policy

```
New-CsApplicationAccessPolicy `
    -Identity "Tag:TeamsMeetingsIntegration" `
    -AppIds "<ApplicationClientID>" `
    -Description "Restricts Teams meeting creation to approved operator accounts"
```

Validate Policy Creation

```
Get-CsApplicationAccessPolicy
```

Record the exact policy identity name.

Step 5: Assign the Application Access Policy

Purpose

The Application Access Policy controls which user accounts the application may act on when creating Teams meetings using application permissions.

Without this policy:

- The application may attempt broader access
- Governance controls are weakened

Assign Policy to Approved Users

Example:

```
Grant-CsApplicationAccessPolicy `
    -Identity user@domain.com `
    -PolicyName "Tag:TeamsMeetingsIntegration"
```

Validate Assignment

```
Get-CsOnlineUser -Identity user@domain.com |
Select UserPrincipalName, ApplicationAccessPolicy
```

> **Important Notes**

- Policies are assigned per-user
- Native group assignment is not supported
- Automation is recommended for larger user populations

Step 6: Testing / Validation

Recommended Safe Rollout Approach

Start with:

- Test tenant
- Small pilot group
- Non-production accounts

Validate Graph Authentication

Example:

```
Connect-MgGraph `
    -TenantId "<TenantID>" `
    -ClientId "<ClientID>" `
```

```
    -CertificateThumbprint "<Thumbprint>"
```

Validate Teams PowerShell Authentication

```
Connect-MicrosoftTeams `
    -TenantId "<TenantID>" `
    -ApplicationId "<ClientID>" `
    -CertificateThumbprint "<Thumbprint>"
```

Test Meeting Creation Using Postman

Token Endpoint

Use:

- OAuth2 Client Credentials flow

Grant type:

```
client_credentials
```

Important Distinction

Ensure testing uses:

- Application permissions

NOT:

- Delegated user authentication

This is critical because:

- Application Access Policies only apply to app-only authentication flows

Test Online Meeting Creation

POST request:

```
POST https://graph.microsoft.com/v1.0/users/{user-id}/onlineMeetings
```

Example payload:

```
{
  "startDateTime": "2026-06-11T10:00:00Z",
```

```
  "endDateTime": "2026-06-11T10:30:00Z",
  "subject": "Teams Integration Test",
  "participants": {},
  "lobbyBypassSettings": {
      "scope": "everyone",
      "isDialInBypassEnabled": true
  },
  "allowedPresenters": "everyone"
}
```

Expected Behaviour

Users WITH policy assignment

- Meeting creation succeeds

Users WITHOUT policy assignment

- Request should fail with authorization-related error

Step 7: Monitoring & Validation

Entra Sign-In Logs

Navigate to:
Entra Admin Center > Monitoring > Sign-in logs

Review:

- Service principal sign-ins
- Authentication failures
- Unusual locations
- Unusual application activity

Audit Logs

Review:

- App permission changes
- Consent grants
- Policy modifications

Teams PowerShell Validation

Validate assigned users:

```
Get-CsOnlineUser |
Where-Object {
    $_.ApplicationAccessPolicy -eq "Tag:TeamsMeetingsIntegration"
}
```

Microsoft Graph Monitoring

Monitor:

- Failed API requests
- Excessive meeting creation
- Abnormal usage patterns

Step 8: Enforcement / Go-Live

Before Production Rollout

Validate:

- Policy assignment scope
- Authentication method
- Third-party application configuration
- Logging and monitoring
- Pilot user testing

Go-Live Activities

- Enable production application configuration
- Assign approved operator accounts
- Confirm successful meeting creation
- Monitor sign-in activity closely for first 7 days

Post-Go-Live Monitoring

Pay close attention to:

- Unexpected meeting creation volume
- Authentication anomalies
- Unauthorized access attempts
- Service principal activity

> ## Important Considerations

Delegated vs Application Authentication

This is one of the most important concepts in this integration.

Delegated Authentication

- User signs in interactively
- User acts as themselves
- Application Access Policy does NOT apply

Application Authentication

- App acts independently
- No user interaction required
- Application Access Policy IS enforced

Improper testing using delegated authentication can lead to incorrect assumptions about policy enforcement.

Cross-Tenant Meeting Access

External tenant users may:

- Require lobby admission
- Experience authentication prompts depending on federation configuration
- Behave differently depending on how the meeting was created

Summary

This implementation enables secure integration between Microsoft Teams and third-party applications using Microsoft Graph OnlineMeetings APIs.

The solution uses:

- Entra ID App Registrations
- Microsoft Graph Application Permissions
- Teams Application Access Policies
- Certificate-based authentication

The configuration provides:

- Controlled Teams meeting automation
- Restriction of application scope to approved users
- Improved governance and operational security
- Reduced tenant-wide exposure in the event of compromise

Proper implementation and testing of Application Access Policies is critical to ensuring the integration operates securely and as intended.

Revision #4
Created 2026-06-11 09:31:57 UTC by AK. Udofeh
Updated 2026-06-11 09:49:44 UTC by AK. Udofeh