Implement Azure security
└ **3.1 Implement user authentication and authorization**
└ **3.1.4 Implement solutions that interact with Microsoft Graph**

1. What is Microsoft Graph and what can it access?
2. How do you register an app to use Microsoft Graph?
3. What permissions are required to access Microsoft Graph?
4. How do you authenticate and call Microsoft Graph using MSAL?
5. How do you read user profile data from Microsoft Graph?
6. How do you list groups or check group membership?
7. How do you call Microsoft Graph from a background service?
8. How do you handle access token scopes and consent?
9. How do you use Graph SDK vs direct REST API?
10. What are best practices for calling Microsoft Graph securely?

---

1. What is Microsoft Graph and what can it access?

Microsoft Graph is a unified API endpoint (graph.microsoft.com) for accessing Microsoft 365 services like Entra ID (users, groups), Outlook, SharePoint, OneDrive, Teams, and more.

---

2. How do you register an app to use Microsoft Graph?

- In Azure Portal → Entra ID → App registrations → New registration
- Add API permissions for Microsoft Graph
- Optionally configure redirect URI and generate client secret or cert

---

3. What permissions are required to access Microsoft Graph?

- Delegated (signed-in user): e.g., User.Read, Mail.Read
- Application (daemon app): e.g., User.Read.All, Group.Read.All
  Some permissions require admin consent.

---

4. How do you authenticate and call Microsoft Graph using MSAL?

Acquire token via MSAL, then use HTTP or SDK.
Example (C#):

```
var result = await app.AcquireTokenForClient(scopes).ExecuteAsync();
var token = result.AccessToken;
```

---

5. How do you read user profile data from Microsoft Graph?

Use GET https://graph.microsoft.com/v1.0/me (delegated)
or GET /users/{id} (application permission).
Include access token in Authorization header:
Authorization: Bearer <token>

---

6. How do you list groups or check group membership?

- List groups: GET /groups
- Check membership: GET /me/memberOf or /users/{id}/memberOf
  Requires permissions like Group.Read.All.

7. How do you call Microsoft Graph from a background service?

Use application permissions with the client credentials flow:

- Acquire token via AcquireTokenForClient()
- Call Graph API using token; no user context needed.

---

8. How do you handle access token scopes and consent?

Scopes define the resources and actions an app can request.

- Delegated: Scopes like User.Read are granted on sign-in.
- Application: Requires admin consent via Azure Portal or admin consent URL.

---

9. How do you use Graph SDK vs direct REST API?

- SDK (e.g., Microsoft.Graph NuGet): Typed clients, fluent syntax, easier integration.
- REST: More control, immediate support for latest endpoints.
  Both use the same access tokens.

---

10. What are best practices for calling Microsoft Graph securely?

- Use least privilege scopes.
- Store secrets in Key Vault.
- Use managed identities if available.
- Validate token claims in APIs.
- Handle token caching and expiration properly.