1. Email Address Synchronization Solutions

A. PowerShell (EmailSyncFix.ps1)

- Authenticates to Exchange Online using admin credentials
- Triggers an immediate delta sync in Azure AD Connect
- Waits 60 seconds for sync completion
- Forces primary email address change for specified user
- Verifies the change was applied
- Cleans up the PowerShell session

Key Technical Aspects:

- Uses Exchange Online PowerShell module
- Requires global admin or Exchange admin rights
- Modifies both proxyAddresses and primarySMTPAddress
- Includes safety delay for sync operations

B. Python (EmailGraphFix.py)

- Implements OAuth 2.0 client credentials flow
- Acquires Microsoft Graph API access token
- Updates both mail and userPrincipalName attributes
- Uses PATCH request for partial user object update
- Requires registered Azure AD app with User.ReadWrite.All permission

2. SharePoint Domain Change Solutions

A. PowerShell (SharePointDomainChange.ps1)

- Establishes admin connection to SharePoint Online
- Executes tenant-level domain name change
- Verifies the root site URL reflects the change

- Automatically disconnects the session

Technical Requirements:

- SharePoint Online Management Shell module
- Global admin or SharePoint admin privileges
- Changes propagate across all site collections

B. PnP PowerShell (SharePointPnPChange.ps1)

- Installs PnP PowerShell module if missing
- Uses interactive authentication (modern auth)
- Provides enhanced SharePoint management capabilities
- Includes automatic module installation

C. C# Azure Function (SharePointMonitor.cs)

- Serverless implementation for monitoring
- Uses timer trigger (runs daily at 6 PM UTC)
- Implements confidential client auth flow
- Leverages Microsoft Graph SDK
- Requires app registration with Sites.Read.All permission

Critical Security Notes:

1. All scripts require proper authentication:
    - Admin credentials for PowerShell
    - Client secret for Python/Graph API
    - App registration for C# solution
2. Permission requirements:
    - Exchange Online: Organization Management role
    - SharePoint: Global admin privileges
    - Graph API: Directory.ReadWrite.All (users), Sites.Read.All (SharePoint)
3. All production implementations should:
    - Store credentials in Azure Key Vault
    - Implement error handling
    - Include logging mechanisms
    - Follow least privilege principle

Execution Requirements:

- PowerShell 5.1+ or PowerShell 7+
- Python 3.6+ with requests library
- .NET Core 3.1+ for Azure Function
- Registered Azure AD applications for API solutions

These solutions provide complete, production-ready implementations for your Office 365 management scenarios while maintaining security best practices