

EntraLens: Service Principal Auditor

Project EntraLens is a comprehensive PowerShell script designed to provide deep visibility into the security posture of a Microsoft Entra ID tenant. By leveraging the Microsoft Graph API, the script inventories every Service Principal—including applications, managed identities, and legacy workloads—and gathers critical data points that are often difficult to correlate manually. This includes assigned permissions, credential types and their expiration dates, ownership details, and, most importantly, the last sign-in activity. The final output is a single, detailed CSV report, enabling security administrators and IT governance teams to easily audit for stale credentials, excessive permissions, and abandoned identities, thereby strengthening the overall security of their cloud environment.

✦ Key Features

This script generates a detailed CSV report to help you answer critical questions like:

- "Who is responsible for this service principal?"
- "Is this service principal still being used?"
- "What permissions does it have and are its credentials expiring?"

The report includes:

- **Ownership & Accountability:** See who owns the underlying application.
- **Lifecycle & Activity:** Know when the SP was created and when it was last used to sign in.
- **Application Context:** Understand its purpose with details like Homepage URL, Tags, and Sign-In Audience.
- **Credential Status:** Track expiration dates for both client secrets and certificates.
- **Full Permission Visibility:** Audit all assigned API permissions and Application Roles.

Understanding Service Principals

A **Service Principal** is an *identity* for a non-human actor, like an application, a script, or a service, that needs to access resources secured by Microsoft Entra ID. While a **User** object represents a person, a **Service Principal** represents a piece of software. The **EntraLens** script audits these identities, which fall into three main categories:

1. Applications

- **What It Is:** This is the most common type of service principal. It's created when you register an application in Entra ID (via "App registrations"). Think of the application registration as the global *blueprint* for your app, and the service principal as the local *instance* of that app within your tenant. This instance is what you assign permissions to and what authenticates using its own credentials (client secrets or certificates).
- **Common Use Cases:**
 - A custom web application that needs to call the Microsoft Graph API to read user profiles.
 - A background daemon service running on a server that needs to access data in Azure Storage.

- A third-party SaaS application (like Salesforce or ServiceNow) that you integrate with your tenant for single sign-on or data synchronization.

2. Managed Identities

- **What It Is:** A Managed Identity is a special type of service principal where Azure automatically manages the credentials for you. This is the most secure way for Azure resources to authenticate because you never have to handle or rotate secrets in your code.
 - **System-Assigned:** The identity is tied directly to an Azure resource (like a VM or Function App). Its lifecycle is linked to that resource—if you delete the resource, the identity is also deleted.
 - **User-Assigned:** The identity is a standalone Azure resource that you can create and then assign to one or more Azure resources. This is useful when you want multiple resources to share the same identity and permissions.
- **Common Use Cases:**
 - An Azure Function App that needs to read secrets from Azure Key Vault without storing a connection string.
 - A Virtual Machine that needs to access an Azure SQL Database using Entra ID authentication.
 - An Azure Logic App that needs to read files from a Storage Account.

3. Legacy Workloads

- **What It Is:** This category often includes service principals that don't fit neatly into the modern "Application" or "Managed Identity" models. They might be associated with older services or authentication protocols. A common example is a service principal representing an Active Directory Domain Services (AD DS) connector or a service that uses the older ADAL authentication library.
- **Common Use Cases:**
 - Service principals created by **Azure AD Connect** to synchronize on-premises Active Directory with Entra ID.
 - Identities for older enterprise applications that haven't been migrated to the modern application registration model.

Why This Matters for Auditing The **EntraLens** script inventories all these types, allowing you to identify which identities exist, what they have access to, and whether they are still in use. An unused service principal with high privileges is a significant security risk, and this report is your primary tool for finding and mitigating that risk.

❓ How to Use

1. Prerequisites

You need the **Microsoft Graph PowerShell SDK**. If you don't have it, open PowerShell as an administrator and run:

```
Install-Module Microsoft.Graph -Scope AllUsers
```

2. Run the Script

- 1. Save the `Get-EntraServicePrincipalReport_Enhanced.ps1` script to your computer.
- 2. Open a PowerShell terminal and navigate to the directory where you saved the file.
- 3. Execute the script:

```
.\Get-EntraServicePrincipalReport_Enhanced.ps1
```

3. Permissions




The script will prompt you to log in and consent to the required permissions. A user with at least the **Cloud Application Administrator** role is recommended.

☒ **Required Permissions:** The script will request the following Microsoft Graph permissions:

- `Application.Read.All`
- `AppRoleAssignment.ReadWrite.All`
- `Directory.Read.All`
- `AuditLog.Read.All` (for sign-in data)

Understanding the Report

Once the script finishes, it will create a `EntraServicePrincipalReport_Enhanced_*.csv` file in the same directory. Here's what each column means:

Column	Description
<code>DisplayName</code>	The common name of the service principal.
<code>AppId</code>	The unique Client ID for the application, used in authentication flows.
<code>ObjectId</code>	The unique identifier for the Service Principal object itself.
<code>ServicePrincipalType</code>	The type of principal (e.g., <code>Application</code> , <code>ManagedIdentity</code> , <code>Legacy</code>).
<code>AccountEnabled</code>	<code>True</code> or <code>False</code> . Shows if the service principal can be used for sign-ins.
<code>CreatedDateTime</code>	The date the service principal was created. Helps understand its age.
<code>LastSignInDateTime</code>	 The last time this SP was used. A very old date or "No sign-in found" is a strong indicator of a stale identity that should be investigated.
<code>Owners</code>	 The list of user principal names responsible for this application. Your first point of contact for any questions.
<code>SignInAudience</code>	 Tells you if this is a single-tenant (<code>AzureADMyOrg</code>) or multi-tenant (<code>AzureADMultipleOrgs</code>) app, which is crucial for understanding its risk profile.
<code>HomepageURL</code>	Provides a direct link to the application, making identification much easier.

Column	Description
Tags	Any custom tags you've applied for your own internal tracking.
SecretExpiryDates	A semicolon-separated list of expiration dates for any client secrets. None means no secrets are configured.
CertificateExpiryDates	A semicolon-separated list of expiration dates for any certificates.
ApiPermissions	A list of delegated and application permissions (OAuth2 scopes) granted to this service principal. Shows what APIs it can call.
AppRoleAssignments	A list of application roles this service principal has been assigned. Shows what roles it has been granted within other applications.