Leen Candidate Exercise: Identity Provider (IDP) Model Extension

Part 1: Enhancing User Object

My approach involved reviewing and researching case studies to understand the process's client-facing applications. In a real-life scenario, I would focus on understanding what prompted the need to enhance specific fields, gather feedback from clients, and gain clarity on the underlying use case.

I used tools like Context 7 and GPT to understand the available endpoints comprehensively. I researched each API endpoint to outline the context and meaning of each field within the specific API structure. Finally, I used Python to normalize the structure, ensuring that data types, indentation, and formatting were properly accounted for.

Github Link for API Endpoint and Final Mappings.

Fields updated:

1. `last_password_changed_at -> password_changed_days.`

Fields added:

1. `mfa_type:"call/email/push/question/signed_nonce/sms/token/token:hardware/`
2. `mfa_status:"ACTIVE/DISABLED/ENROLLED/EXPIRED/INACTIVE/NOT_SETUP/PENDING_ACTIVATION"`
3. `mfa_providoer: MS Entra/Okta`
4. `Access_levels: (only for Okta) custom, from vendor`
5. `Assigned_role: custom, from vendor`

Gaps:

1. MS Entra does not have a standardized method of defining user `access levels`; however, depending on client requirements, a field can be added to return 'groups' for the user.
2. To derive enough context, it requires querying multiple API endpoints for each user, such as for MS Entra, using $select to get the status of the user account and the last login date.
3. `Vendor_created_at, activated_at, last_status_changed_at, last_updated_at, and password_changed_days` are not populated for data from MS Entra, due to limited data granularity from endpoints.

API Endpoints:

1. Users API (Standard API to list all Users for an IDP)
2. List User Factors (supports deeper understanding of MFA status, type, and enforcement)
3. List User Role Assignments (supports deeper understanding of the roles granted to the user)
   a. Roles overview (Documentation for roles in Okta)

The following API Endpoints were queried for MS Entra ID:

1. List all Users
2. Get User Registration Details

Part 2: Creating a Policy Object

To create a unified policy object for Leen, I followed a structured approach similar to Part 1, identifying and querying relevant endpoints from Okta and Microsoft Entra. Okta has a significantly more granular policy API than MS Entra, with well-defined schemas for each 'type' of policy (e.g., passwords, MFA, etc). Approach 1 was centered around having a well-defined schema, with pre-defined policies in Leen's object. This approach was abandoned as it limited the flexibility of extracting polices that may fall outside the pre-defined scope and required significantly more computation to extract and map policies from MS Entra. This also reduced the flexibility of the data extraction process.

Approach 2 creates a standardized Leen object optimized for universal flexibility. The object has standardized fields like status, data_created, and policy_type, and a 'details' field that captures information unique to each policy, such as overrides, specifications, etc.

Github Link to final JSON.

Gaps:

1. Unless specified otherwise, there is less control over what 'type' of policies are queried. For instance, this method would return all policies defined by the organization.
2. MS Entra does not populate the `priority` and `type` for policies.

Potential Future Solution:

1. Github Link: here

Understanding Identity Providers:

1. What is an identity provider (IdP)? | Cloudflare
2. What Is Identity Provider (IdP) Security? | CrowdStrike
3. Applications of identifying IdPs

Challenges Using Identity Providers

1. Guardians of the Clouds: When Identity Providers Fail

Understanding Client-Facing Applications for User Object:

1. Reason for request? Understanding the existing gap/need to request the field?
2. Case Study: Streamlining Access Reviews

Understanding Policies:

1. List all policies in the organization: https://learn.microsoft.com/en-us/graph/api/conditionalaccessroot-list-policies?view=graph-rest-1.0&tabs=http
2. For MFA: https://learn.microsoft.com/en-us/graph/api/authenticationmethodspolicy-get?view=graph-rest-beta&tabs=http

3. Passwords:https://learn.microsoft.com/en-us/graph/api/passwordauthenticationmethod-get?view=graph-rest-1.0&tabs=http
4. SignIn: https://learn.microsoft.com/en-us/graph/api/signin-get?view=graph-rest-1.0&tabs=http
5. Access:https://learn.microsoft.com/en-us/graph/api/resources/appliedconditionalaccesspolicy?view=graph-rest-1.0#properties

-