SCIM Delta Query - Update

Anjali Sehgal (AWS) and Danny Zollner (Microsoft)

Context

Goal

- Enable ***incremental retrieval of resources*** that have been created, updated or deleted in a SCIM service provider since the last request.
- This allows for more efficient interactions between SCIM clients and service providers and addresses problems that have inhibited ***large-scale*** implementation of use cases such as ***synchronization***, ***entropy detection***.

Why is it Important?

- Potential synchronization inaccuracies could lead to data divergence between the SCIM client and SCIM service provider. Undetected diverging data between a SCIM client and SCIM service provider can lead to undesirable authorization decisions.
- End to end reconciliation processes, reduces the risk of incorrect authorization decisions based on divergent states between client and server

This data divergence detection may be used for r**eporting purposes** or may be extended to either **trigger provisioning** of those resources **into the target system** or **pulling changes from the target system** into the source.

What are we working on?

We are working on simplifying and optimizing the solution further.

Some of the aspects we are looking at

1. Full Scans
    a. Should delta query return full data set as part of initial delta token retrieval or should delta query only focus on returning changed (created, modified, deleted) entities only?
    b. Decoupling acquisition of an initial delta token from any other requests makes the most sense
2. Delta Query Response
    a. How can we optimize the response?
        i. Return full resource state for modified resources
        ii. Return modified attributes only for modified resources.

Full Scans

Possible Flow

Step1: Obtain the first Delta Token without full scan: SCIM Client calls delta query without a token. The Delta Query responds with a **nextDeltaToken**, however does not return any resources as part of this call.

Step2: Client Completes a Full Scan using List Users/List Groups APIs

Step3: Using Delta Token to perform a Delta Scan: SCIM Client executes the delta query after completing the full scan using the delta token issues by the server as part of response from initial delta query from step 1.

Benefits

- Simplifies the Delta Query
- SCIM clients can continue to use existing APIs to do full scans leveraging their existing integrations
- Delta Query will be an additional addon to optimize their existing implementations.

Delta Query Response

We are looking at possibilities to optimizing Delta Query response even further.

- Possibility to return ***changed attributes only*** as an option in **addition to the option** of returning ***full resource state*** for the modified resource.

Goal

- Widely adoptable by SCIM servers sourcing data from common databases/data sources
- Efficient at large scale from server resource consumption perspective.
- Efficient use of network bandwidth (avoid unnecessary data being returned)
- Efficient for client to understand response from server needed to build entropy detection logic

Challenges

- Define a common response schema that allows for retrieval of full resource state (approach 1) and retrieval of modified attributes only (approach 2) to support above goals.
- Single value attributes are easy to handle, however multi-valued attributes pose some complications on how to represent deltas.

Discussion Avenues

We are looking for more members to review and contribute. Please collaborate with us via PRs, Issues etc.

https://github.com/ietf-scim-wg/draft-seh

Or Use QR Code to open the link