TIER API Security Guidelines

Fundamentals of API Authorization

Regardless of the particular machinery involved, users and clients must identify themselves to a service. This means users and clients have identities and credentials.

Initial axioms (from Jim Fox)

1. Internet2 has a recommended identity solution for people users --- Shibboleth and InCommon metadata.
2. Internet2 has an identity system for services --- InCommon Certificate Service.
3. Internet2 is all about federation and inter-institutional collaboration.

Users are already well taken care of with Shibboleth, InCommon metadata and person registries. We are developing APIs for the latter. A service can easily identify a user and know about her (via attributes). Federation allows services to identify users from different institutions.

What is needed is similar support for API clients---a registry of entities. An Entity Registry, of both services and clients, supported by a standardized API, seems necessary. Entity attributes, while not the same as those of a person, are similar and could be handled by similar APIs.

The InCommon Certificate Authority already gives us one potential method of support for entity federation. A client could use its certificate to register with an entity registry, or to get a credential from an authorization service.

Emerging Issues

IssueID	Issue Title	Notes
1	Entities as Agents (Clients, Services) A: Must have a registry in which they are an entry B: Must have accounts/credential sets C: Must be discoverable by potential clients D: Must have a trust anchor	API Security turns out to be the driver for taking up non-person entities
2	Authorization policies have a fundamental structure A: SUBJECT can perform ACTION on RESOURCE under CONDITIONS B: True = Allow C: False = Deny	
3		
4		