Cedar Profile of Oauth 2.0 Rich Authorization Requests

Dean Saxe and Sarah Cecchetti

Amazon

What is Cedar?

- Open source domain-specific language for authorization
- Released in 2023
- Evaluation engines available in Rust and Go
- Evaluation engines are tested against Lean code to prove certain properties of the language using formal methods

Primary Use Case

- Identical to RAR: Client needs to access a protected resource with high granularity. AS facilitates this transaction by “blessing” a Cedar policy set which the client can then communicate to the RS via OAuth token.
- Benefits:
  - RS can use standard Cedar libraries to do policy evaluation locally, reducing latency and increasing flexibility
  - Same functionality as JSON but it provides the AS and RS a common language to improve efficiency and reduces complexity within the RS

Open Questions

- Should an AS be allowed to respond to a non-Cedar formatted request in Cedar format? Or vice versa?
  - We think “no” for the sake of simplicity
- Should we provide guidance about when to use an array in “type” to indicate both intent and format?
- Should this profile be extensible to multiple policy languages?
  - We think “no;” in more complicated use cases, Cedar syntax has unique information like schemas and resource lists that would make it difficult to both extend functionality and maintain a high level of interoperability if multiple languages are involved. Each language should have its own profile.
- We don’t see any security or privacy considerations that surface by using Cedar with RAR that aren’t already present in RAR; are we missing anything?

Appendix: Potential Future Use Case

- What can Alice access? Client needs to know what resources are available to the user.
- Cedar uses a functionality called “partial evaluation” to accomplish this. Rather than requesting a resource, the client designates the resource as “unknown,” the AS returns a list of resources (or a database query to lookup resources), and the client can then send fully formed RAR requests for the resources it needs, and receive a RAR token in return.