Jonathan Roland

WEB 420 RESTful APIs

Instructor: Prof. Krasso

12/04//2020

Discussion 8.1

* **Describe what “Cache-Control” is and how it is used.**

Cache-Control is an HTTP response header that assists server performance by discouraging clients from making pointless requests. It does this by telling the client that if it should want to make the same request for the same resource, to instead look at the currently cached version of that resource. This header can invoke the max-age directive to tell the client how long to wait (in milliseconds) before making the request again. Alternatively, it can also invoke the no-cache directive to represent a resource that is unlikely to be the same even with a very small gap of time between requests.

* **List and describe the available options for managing security in a REST API.**

Generally, security is handled by authentication. First a user establishes credentials, then the credentials are automatically transmitted along with every request. The common options for transmitting those credentials are basic HTTP authentication, OAuth 1.0 and OAuth2.0.

Basic

The basic standard is to encrypt a username/password pair scrambled in Base64. The problem is that any network listener can intercept and decode it, resulting in credential theft. While weak, it can be used safely via HTTPS due to SSL encryption. Essentially, it is a weak security scheme that can be wrapped in a much tougher shell.

OAuth1.0

OAuth is an open standard protocol for delegated access control. This means a client can allow another client or application to use their access credentials without sharing their security details like username and password. This is done when the primary client sets up access for a secondary client through an access server. The secondary client then refers to the access server when making requests for secure resources, rather than passing credentials directly.

OAuth 2.0

OAuth 2.0 is a separate standard from 1.0 but is based on similar principles. This standard seeks to enhance certain considerations over the 1.0 model. These include a more thorough signature matching protocol between client and server, non-browser application support, role specificity, token expiration and token refreshing.

**References:**

Richardson, L., Amundsen, M., & Ruby, S. (2013). *RESTful Web APIs: Services for a Changing World* (1st ed.). O’Reilly Media.

Gamage, S. (2019, March 27). *OAuth 2.0 for Dummies - Sominda Gamage*. Medium. https://medium.com/@somindagamage/oauth-2-0-for-dummies-8807cff5c3b2

Ramchandani, P. (2018, June 13). *What’s the difference between OAuth 1.0 and OAuth 2.0?* Packt Hub. https://hub.packtpub.com/what-is-the-difference-between-oauth-1-0-and-2-0/