Spring Security ve OAuth

harezmi bilişim çözümleri

OAuth Nedir?

- Web üzerindeki değişik servislere, bu servislerin kullanıcıya ait diğer servislerdeki bilgilere asıl username/password bilgilerini vermeden erişim sağlar
- Erişimi belirli ölçütlerle sınırlandırmaya da yarar
  - Zaman, scope vb.
- Sadece HTTP(s) resource'larına erişim sağlar

Oauth Nasıl Ortaya Çıktı?

- Kökeninde OpenID yatmaktadır
- Password paylaşımı ve interaktif login mekanizmasına ihtiyaç duymadan, OpenID gibi servislere API vasıtası ile erişim sağlayacak bir standart ihtiyacından çıkmıştır
  - RFC 5849 – OAuth 1.0
- Mobil araçlar, desktop uygulamalar, web uygulamaları ve web servisleri için built-in desteğe sahiptir

Oauth Neye Benzer?

- Bazı araçlardaki vale anahtarına (valet key) benzetilebilir
- Vale anahtarı araca sınırlı erişim sağlar
- Vale bu anahtası sadece aracı park etmek için kullanabilir

Oauth Örnek ve Terminoloji

- Örneğin bir web kullanıcısı, başka bir yerdeki servis tarafından yönetilen fotoğraflarına, bir print servisin erişmesini sağlayabilir
  - Web kullanıcısı: resource owner
  - Fotoğraf: protected resource
  - Foto servisi: server
  - Print servisi: client

Oauth Terminoloji

- Burada client, resource owner yerine, server'daki bir resource'a erişmek istemektedir
- Bunun için öncelikle resource owner'dan izin almalıdır
  - İzin: token + matching shared secret

Oauth Terminoloji

- 2-Legged, 3-Legged, N-Legged
  - Kimliklendirmede yer alan birimlerin sayısını temsil eder
  - 2-Legged: client-server
  - 3-Legged: resource owner-client-server
  - N-Legged: resource owner – client x N - server
- N-Legged: Erişim izninin diğer client'larla paylaşıldığını/delege edildiğini ifade eder

Oauth Terminoloji

- Credential bilgisi üç çeşittir (parantez içi spec'deki isimlendirmedir)
  - Client credential (consumer key + secret)
  - Temp credential (request token + secret)
  - Token credential (access token + secret)
- Client credential: istemcinin kimliklendirilmesinde kullanılır
- Token credential: resource owner'ın username + password'ü yerine geçer

Oauth'un İşleyişi

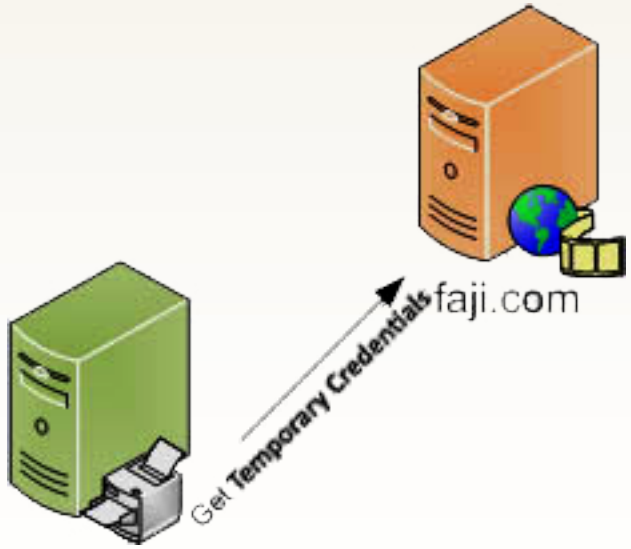
- Jane, faji.com sitesine kendine ait fotoğraflarını upload etmiş olsun

Oauth'un İşleyişi

- Bazı fotoğraflarını da internet üzerinden print hizmeti veren beppa.com ile bastırmak istesin
- Bu durumda Beppa, fiji.com ile entegre olurken Fiji.com'un Oauth mekanizmasında kullanmak üzere client credential + secret key bilgisi edinir

Oauth’un İşleyişi

- Jane, beppa.com'da print etme işlemini başlattığı anda, beppa.com faji.com'dan temp credential ister



Temp credential, resource owner'a özel değildir, protected resource'lara erişim için resource owner'dan onay almak için kullanılır

Oauth'un İşleyişi

- Jane, onay için beppa.com'dan faji.com'a yönlendirilir

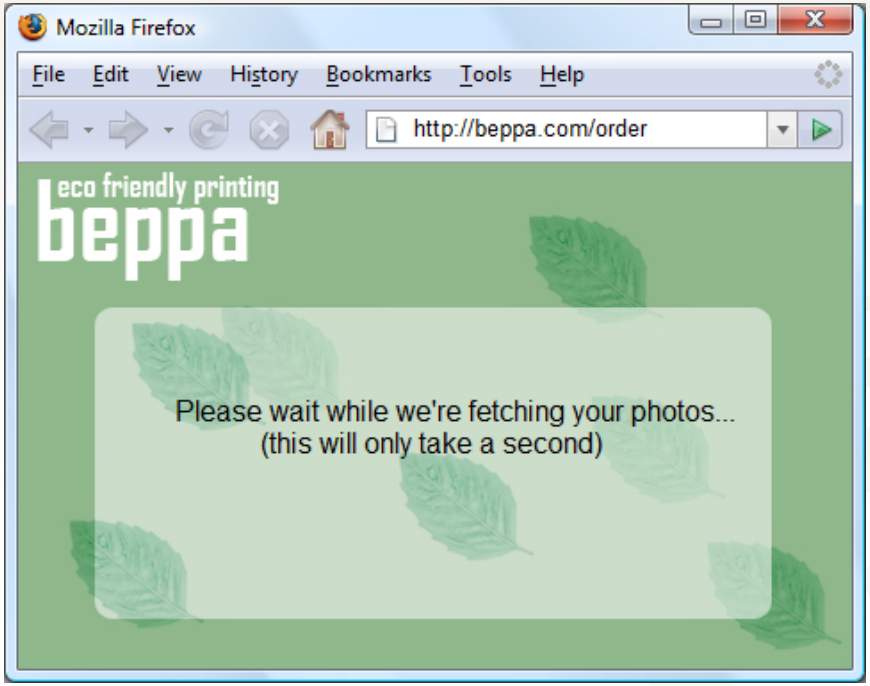
Oauth'un İşleyişi

- Onay verdikten sonra tekrar beppa.com'un belirli bir adresine yönlendirilecektir

Onayla birlikte beppa.com'un temp credential'ı geçici bir süre Kenan'ın protected resource'larına erişim yetkisine sahiptir

Oauth'un İşleyişi

- Beppa.com onay aldıktan sonra, temp credential (request token) kullanarak access token elde eder



Oauth'un İşleyişi

- Beppa.com artık resimlere erişmiştir. Print işlemine başlanabilir

Oauth’un İşleyişi

- Qauth istekleri sayısal imzalamaya tabi tutulur
- Request'i gönderecek taraf request'in hash'ini tespit eder
- Bu hash verisi “shared secret key” ile şifrelenir

Oauth’un İşleyişi

- Request'i alan taraf önce request'in hash'ini oluşturur,
- Daha sonra kendine gönderilen şifreli hash'i elindeki “shared secret key” ile çözerek bu hash ile karşılaştırır
- Değerler aynı ise istek değişmeden ulaşmış demektir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com