XML Consortium

～ 第7回 XMLコンソーシアムWeek ～

WebOS間連携における認証方法の検討状況
OAuthの利用

２００８年６月６日

25min

aPlat プロジェクト

アドソル日進株式会社　荒本道隆

OAuthとは

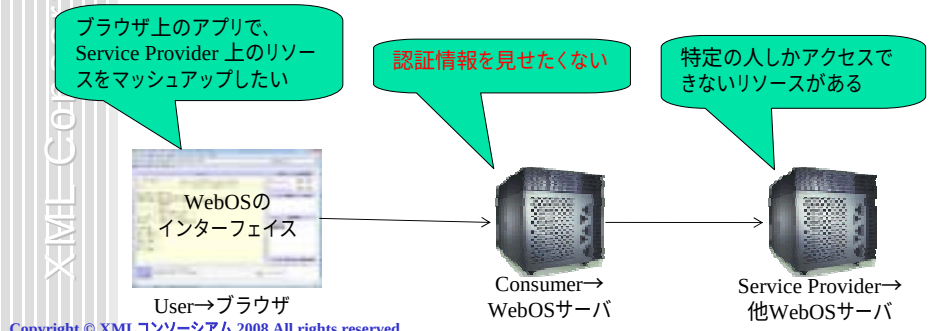
XML Consortium

Webサイトやアプリケーションが、ユーザがWebサービスの認証情報をWebサイトやアプリケーションに開示することなく、APIを通じてWebサービスから保護されたリソースにアクセスするためのプロトコル（抄訳）

第7回XMLコンソーシアムWeek
セキュリティ部会「認証技術調査の最新成果」より引用

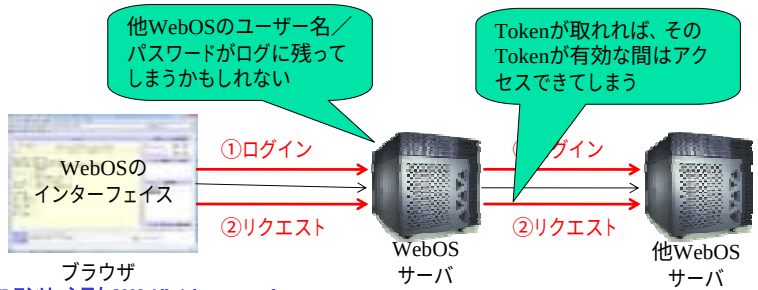
OAuth概要

- 用語説明
  - User：サイト利用者→ブラウザ
  - Consumer：Userが利用しているWebサイト→WebOSのサーバ
  - Service Provider：Webサービス→他WebOSのサーバ



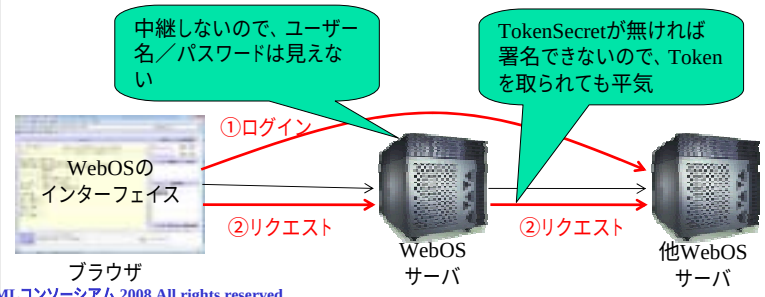
今までの他WebOSへのアクセス

- WebOSから他WebOSや他サイトにアクセスする場合
  - WebOSサーバで中継する必要がある
    - JavaScriptのドメイン境界の制限のため
  - ログインも中継する
    - 他WebOSのユーザー名／パスワードが、サーバ側で見えてしまう
  - リクエストの中に、毎回Tokenが入っている
    - そのTokenさえあれば、ニセのリクエストを発行できてしまう



OAuthを使った場合

- OAuthを使って、他WebOSや他サイトにアクセスする場合
  - WebOSサーバで中継する必要がある
    - JavaScriptのドメイン境界の制限は変わらない
  - ログインは他WebOSサーバに直接行う
    - WebOSサーバでは、中継しないので何もできない
  - リクエストの中に、TokenSecretは入れない
    - 署名にTokenSecretを使っているので、ニセのリクエストは発行できない



なぜ、OAuth？－1

- メリット
  - マッシュアップでは、アクセス権がもらえれれば十分な場合が多い
  - 事前に鍵（Consumer Key, Consumer Secret）と署名方式を交換しておくだけ
  - CookieやJavaScriptなど、ブラウザ特有の機能に依存しない
    - ブラウザ以外からも利用しやすい
  - 通信内容がシンプル
    - 3つのメソッドを呼び出すだけ
    - HTTPが使えて、署名ができること
  - サンプルコードが豊富
    - C#, ColdFusion, Objective-C, Java, **JavaScript**, Jifty, Perl, PHP, Python, Ruby
      http://oauth.net/code

なぜ、OAuth？－2

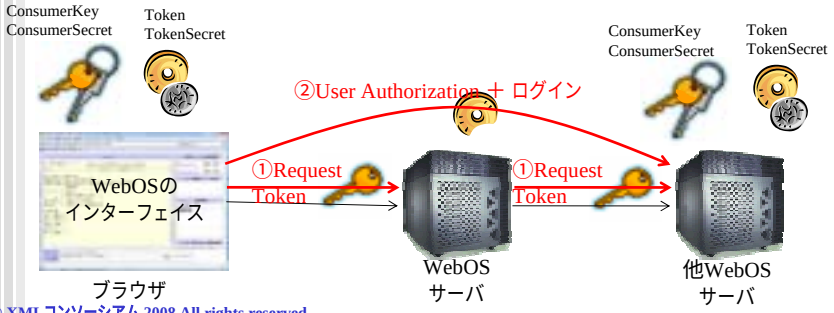
- デメリット
  - 一度だけ、WebOSサーバ上をToken, TolenSecret がセットで通過する
    - 認証情報を通過させないことが、OAuthの目的
  - 鍵（Consumer Key, Consumer Secret）を交換しておかなければならない
    - Service Providerとの個別対応が発生
    - マッシュアップでは、不特定多数のService Providerとやり取りしたい

WebOSでのOAuthの流れ－1

0．事前に、ConsumerKey, ConsumerSecretを交換しておく
- 交換方法については、OAuthの範囲外

1．Request Token → Token, TokenSecretを取得する
- ConsumerSecretを使って署名

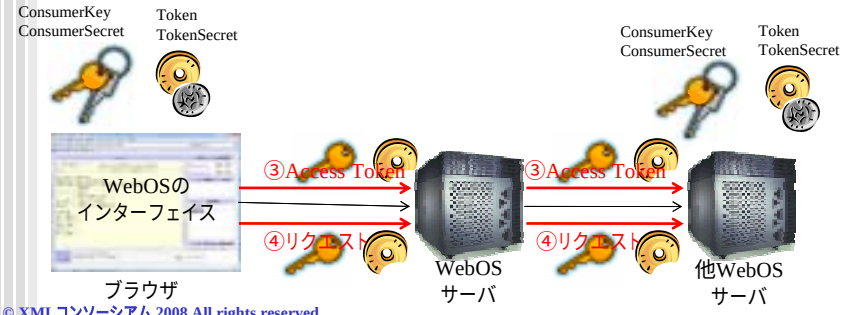
2．User Authorization → Token とアカウントを対応させる
- 他WebOS側のユーザー名／パスワード入力は、直接アクセス



WebOSでのOAuthの流れ－2

3．Access Token → Token, TokenSecretを取得する
- ConsumerSecretとTokenSecretを使って署名

4．目的のリクエストを発行
- ConsumerSecretとTokenSecretを使って署名



WebOSで起こりえる課題－1

- 中継しているように見える
  - WebOS上のブラウザ内で表示すると、中継しているように見える
  - ユーザー名／パスワードが見られているか分からない
  - 解決方法：許可をおこなうための操作は、別窓・別タブで行う

YouOSで、他サイトを表示した例

別タブで、他サイトを表示した例

WebOSで起こりえる課題－2

- Tokenの有効期限切れ
  - Tokenの有効期限が切れると、Tokenが無効になる
  - 解決方法：リトライ処理を実装する？
  - タイムアウトするたびにログインして再許可を行うのは面倒
    - Service Provider へのログインは、Cookieなどを使って省略可能にする
    - Service Provider に依存するので、今回は対応しない

WebOSでのOAuth対応

- WebOSで、OAuth対応アプリを作れるか？
  - 一部のWebOSが提供している開発機能
    - 他サイトへの中継
    - JavaScriptによる独自アプリの実装 ← OAuthにサンプルあり

↓

提供されている開発機能で、OAuth対応アプリは作れそう

↓

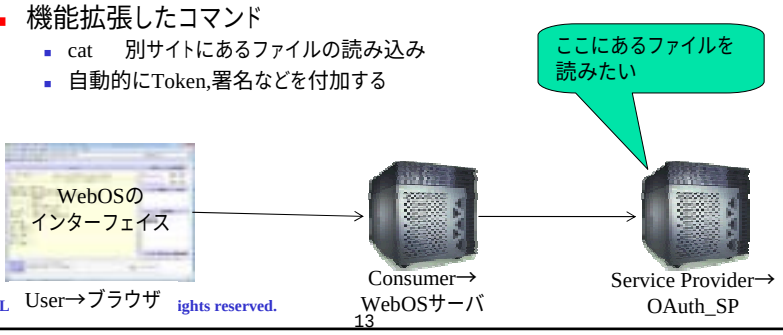
全アプリに、わざわざOAuth対応のコードを入れるのは大変

↓

WebOSのAPIでOAuth対応すれば、全アプリが対応した事に

WebOSでOAuthを使ったデモ

- WebShell（仮称）
  - Webサービス実証部会で実証実験のために開発したWebOS
- OAuth対応方法
  - OAuth のためのコマンドを追加
    - oauth_RequestToken
    - oauth_UserAuthorization
    - oauth_AccessToken
  - 機能拡張したコマンド
    - cat　別サイトにあるファイルの読み込み
    - 自動的にToken,署名などを付加する



実装してみて

- HTTPの通信内容を見れば、問題が解決できる
  - 署名以外は、HTTPを見ればすぐに分かる
- JavaScriptのサンプルは、とても役に立った
  - ほとんどそのまま貼り付けただけ
  - 作業時間：半日
- Authorizationでアクセスした時のレスポンスが、それぞれ違う
  - HTMLが返ってきて、それを人が見て、「許可」を選択する
- OAuthに対応したサービスはまだまだ少ない
  - OAuthに対応したWebOSはまだ存在しない
    - Consumerとなるのか？ Service Providerとなるのか？
  - Ma.gnolia
    - ConsumerKey, ConsumerSecret を交換する手順が分からなかった
  - Twitterは残念ながら試せなかった
    - http://twitter.com/oauth が停止中（2008/05/26）

今後の展望編

に続きます