3Y-8

OAuthにおけるトークンを用いた権限管理方式の提案と実装

後藤 浩行† 武 佑香† 鳥居 悟‡ 齋藤 孝道†
明治大学† 株式会社富士通研究所‡

1 はじめに

インターネットの普及とともに Web を通してサービスを提供する Web アプリケーションが広く普及している．更に，近年では，運営主体の異なる複数の Web アプリケーションを組み合わせてサービスを提供するマッシュアップサービスが行われるようになった．

従来，マッシュアップサービスが，Web アプリケーションがそれぞれ保持しているエンドユーザ情報を，エンドユーザに代わり，利用する場合，マッシュアップサービス側でエンドユーザに ID とパスワードを保持する運用形態が多かった．それに対し，エンドユーザの持つ情報にアクセスする権限を，マッシュアップサービスなどを含めた Web アプリケーションに安全に委譲する仕組みとして OAuth［1］が提案された．

本論文ではその OAuth を用いたアクセス制御の一例として，LMS （Learning Management System）である Moodle［2］という Web アプリケーションにおける権限委譲方式の実装を示す．

2 OAuth

2．1 概要

OAuth とはエンドユーザの権限の委譲を実現するプロトコルである．OAuth1.0a では電子署名を行っていたが OAuth2.0 では HTTPS（Hypertext Transfer Protocol over Secure Socket Layer）を利用する．OAuth2.0（以降，区別しない場合は OAuth と呼ぶ）では，EndUser のリソースを保持する ResourceServer が EndUser の同意のもと Client に権限を委譲することが出来る．この際，EndUser の ID，パスワードといったクレデンシャルを渡さ無いことが特徴である．

2．2 関連用語

ここで，OAuth に関する用語を説明する（図１）．

● アクセストークン
アクセストークンは，作成時に都度生成されるランダムな文字列により，特定される情報であり AuthorizationServer（後述）に保存される．作成時，権限の委譲範囲を示す Scope，有効期限を示す Expires，任意のオプションを合わせて保存される．
EndUser（後述）の承認があった際，後述の Client にアクセストークンの作成を行う．

● ResourceServer
EndUser（後述）のリソースを保持し，Client（後述）からの要求時にアクセストークンの検証を行い保護されたリソースの提供を行う．

● EndUser
ResourceServer に保護されたリソースの利用権限を持ち，Client（後述）に保護されたリソースへのアクセスを許可する．

● Client
EndUser から権限の委譲を受けるために，AuthorizationServer にアクセストークンの要求をする．そのアクセストークンで対応する ResourceServer 上のリソースにアクセスし，Web サービスを提供する．

2．3 フローの説明．

OAuth2.0 でフローが複数定義されているが，本論文で利用する WebServer フローについて説明する（図１）．

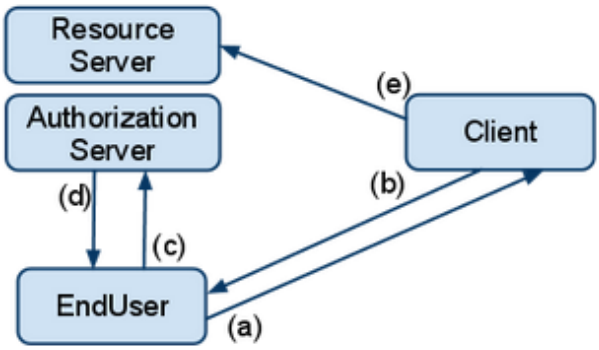

図１.WebServer フロー

a) EndUser が Client にアクセスする．
b) Client により EndUser は AuthorizationServer にリダイレクトされる．
c) EndUser は権限の委譲の可否を伝える．承認時に次に進む．
d) AuthorizationServer はアクセストークンを作成し，（アクセストークン情報を付加し

An Implementation of Access Control Mechanisms using OAuth Access Token
† Hiroyuki Goto,Yuka Take, Takamichi Saito
‡ Satoru Torii
Meiji University(†), FUJITSU LABORATORIES LTD.(‡)

た）リダイレクト先 URL に EndUser は Client にリダイレクトされる.

e) Client は当該アクセストークンに対応したリソースにアクセスする. ここでのリソースとは, ResourceServer の持つ EndUser の保護されたリソースである.

3 Moodle について

授業用の Web ページを作成し資料などを提供するため Web アプリケーションで, 教師が学習用ページを作成し生徒が利用する. 管理者, 教師, 学生などのロールがあり, ロールベースアクセス制御が可能である.

表1 ロールとその権限の例

	管理者	教師	生徒
ユーザ作成	可能	不可能	不可能
コース作成		可能	不可能
コース履修			可能

4 提案システム

4．1 概要

本論文では, AuthorizationServer 上で Moodle におけるロール情報を Client の権限と対応付けるために, アクセストークンの Scope にロール情報を格納するようにした. これにより, Moodle において, OAuth を用いたエンドユーザの権限委譲の仕組みを提案し実装した.

4．2 構成

ここでは, 前述の主体の実現環境を示す.

AuthorizationServer, ResourceServer

- Apache Version 2.2.3
- PHP Version 5.3.3
- MySQL Version 5.1.53
- oauth2-php revision 21

PHP で実装された OAuth2.0 ライブラリである oauth2-php［3］がある. 今回, 改修した oauth2-php を Moodle に組み込んだ.

oauth2-php の改修点は以下の3点である.

- EndUser から渡されるセッション情報を元に, EndUser のロール情報を取得できるようにした.
- アクセストークンを作成しそれを, 保存する際に EndUser の Moodle におけるロール情報と併せて保存するようにした.
- Client が EndUser の保護されたリソースにアクセスする際, アクセストークンと併せて保存した EndUser のロール情報を元にアクセストークンの検証を行うようにした.

Client

- Apache Version 2.2.3
- PHP Version 5.16

oauth2 draft-10 の仕様にあわせて OAuth2.0 のメッセージを送信できる簡易的な Web アプリケーションを新たに実装した.

4．3 動作

提案システムの動作を説明する（図2）.

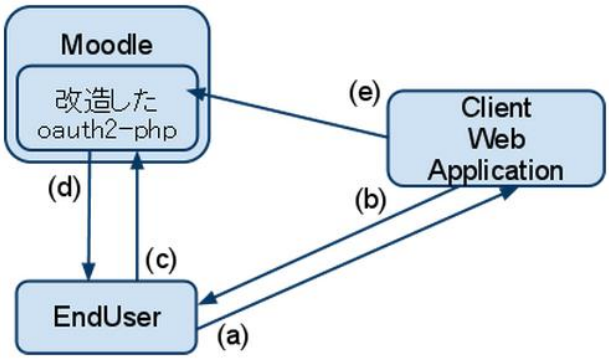

図2. 提案システムの動作

a) EndUser が Client へアクセスする.

b) Client が要求する権限の範囲を URL に付加して EndUser を Moodle（Authorization Server）にリダイレクトさせる.

c) EndUser が権限委譲を承認した際, EndUser により Moodle のセッション情報をクッキーに含めアクセスする.

d) Moodle(AuthorizationServer)はアクセストークンを作成する. この時, EndUser から渡された Moodle のセッション情報を元にログインしているユーザ名, 付与されているロールを取得し, アクセストークンと併せて保存する. リダイレクト先 URL にアクセストークン情報を付加させて EndUser を Client にリダイレクトさせる.

e) Client はアクセストークンを用いて Moodle(AuthorizationServer)上で対応する EndUser の保護されたリソースにアクセスする. Moodle のサービスを利用する際は, アクセストークンにより特定されるロール情報に合わせてアクセス制御を行う.

5 まとめ

アクセストークンの持つ Scope 情報に EndUser が AurhorizationServer に持つロール情報を用いて, Client のロールベースアクセス制御を行い, その実装を示した.

6 参考文献

［1］http://tools.ietf.org/html/draft-ietf-oauth-v2-10
［2］http://moodle.org/
［3］http://code.google.com/p/oauth2-php/