京都産業大学コンピュータ理工学部でのUPKIの取り組み

京都産業大学

POWER UNIV.

実証実験参加の目的

京都産業大学コンピュータ理工学部では JA-SIG で開発されている CAS（Central Authentication Service）により教職員および学生の認証サービスを提供している。UPKIにおいて提案されているShibbolethによる認証連携が今後大学間での標準的な認証の基盤として利用可能かどうかを見極めるため、また、実際にShibbolethによりキャンパス情報システムにおけるサービスを提供する手順を調査するために実証実験に参加し、参加者との技術情報交換およびノウハウの蓄積を目的とする。

2008年度実施内容

Shibboleth IdP, SP, DS の構築

テスト用の Shibboleth Identity Provider, Service Provider, Discovery Service を構築する。

軽量アプリケーションでの Shibboleth 対応手順の調査

Wiki や Ruby on Rails のような Web アプリケーションフレームワークで開発される軽量なアプリケーションにおいて Shibboleth 認証に対応するための手順を調査し、大学における教育研究システムにおいて認証基盤を簡易に利用する方法を検討する。

以下では軽量アプリケーションの調査状況について述べる。

軽量アプリケーションでのShibboleth対応手順の調査

Wikiでのアクセス制御

RubyでのWiki実装HikiにおいてShibboleth SPを用いたアクセス制御方法について検討した。

Shibboleth（mod_shib）でのアクセス制御の問題点

mod_shibでは図１のようにApache設定ファイルにおける<Location>, <LocationMatch>などの「ウェブ空間コンテナ」を用いてShibboleth認証を要求するURLを指定する。

POSTデータ消失問題： Wikiにおいてページ更新に対してアクセス制御をかけたい場合図１のようにPOST, PUT, DELETEなどのHTTPメソッドに対して認証をかける方法が考えられるが、Shibbolethでは認証時にリダイレクトが発生するので、初めてページを更新する際にはPOSTデータが消失してしまう。入力Formを返すURLにも認証をかける必要がある。

URL Query String問題： 「ウェブ空間コンテナ」はURL Query Stringを認識しないため、図２のHikiの例ようにQuery Parameterで動作を指定するCGIには適用できない。

軽量アプリケーションでのShibboleth対応手順の調査

```
<Location "/hiki">
 <Limit POST PUT DELETE>
   AuthType shibboleth
   ShibRequireSession On
   require valid-user
 </Limit>
</Location>
```

図１：Wikiでのアクセス制御の例

```
http://wiki.example.com/hiki/?c=edit;p=FrontPage
```

図２：ウェブ空間コンテナで指定できないURLの例

軽量アプリケーションでのShibboleth対応手順の調査

Ruby on Railsでのアクセス制御

Ruby on Rails（以下Rails）とはプログラミング言語 Ruby を用いて MVC (Model View Controller) アーキテクチャに基づき Web アプリケーションを構築するためのフレームワークである。軽量アプリケーションの開発において採用されることが多い。Railsで実装された redmine というプロジェクト管理アプリケーションのShibboleth対応について調査した。

Railsの環境構築とShibboleth認証の導入方法

図３に示すようにApache2をロードバランサとし、mongrelでRailsを動作させるケースについて調査した。Shibboleth認証についてはApache2に mod_shib を導入し、図４のようにHTTPヘッダで属性情報を受け渡しすることで導入できる。

軽量アプリケーションでのShibboleth対応手順の調査

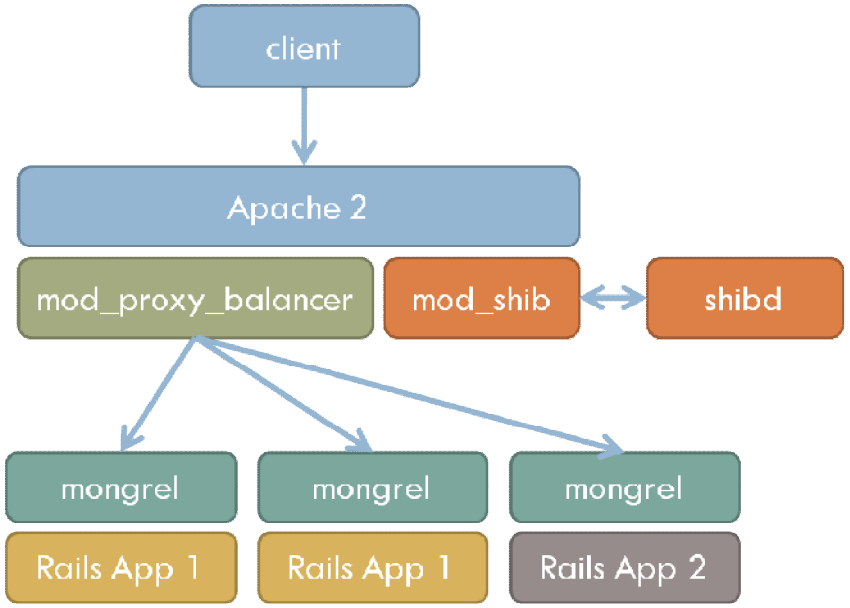

図３：Rails環境の構成例

```
<Location "/rails-app">
 AuthType shibboleth
 ShibRequireSession On
 ShibUseHeaders On
 require valid-user
</Location>
```

図４：Apache2をProxy利用する場合のmod_shibの設定例

軽量アプリケーションでのShibboleth対応手順の調査

Railsでの認証方式の調査

Railsの認証ではrestful-authenticationプラグインなどが良くコントローラのアクセス制御に用いられているが、redmineではさらにモデルのアクセス制御も実装している。
redmine の外部認証機能を拡張してShibboleth対応を行った。

POSTデータ消失問題への対応

Railsでの認証処理はコントローラの action に対して before_filter により設定できる。そのためPOST、PUT、DELETEなどの処理を実行する action だけでなく、入力 form を返す action でも認証を設定すれば問題は回避できる。

軽量アプリケーションでのShibboleth対応手順の調査

Shibboleth対応コードの公開

redmine の 0.8-stable の r2252 を Shibboleth 対応したものを公開している。以下のコマンドでコードの取得およびオリジナルとの比較ができる。

```
% git clone git://gitosis.kyoto-su.ac.jp/redmine.git
% cd redmine
% git diff origin/base-svn master
```

軽量アプリケーションでのShibboleth対応手順の調査

プライバシを考慮したID受け渡し

軽量アプリケーションのサーバは研究目的で一時的に運用されるサーバなどが多く、また、継続運用される場合も運用などがうまく引き継がれず、最悪の場合外部から侵入されてデータが漏洩する危険性がある。このような危険性を避けるためにシングルサインオンの連携範囲を限定するのは、教育研究活動支援の目的に逆行することになる。そこで Shibboleth IdP 側の個人情報を保護しながら連携する方法について調査した。

- ☑ transient-id による連携

Shibbolethのデフォルトの設定で利用できる。毎回異なるIDを受け渡すため、SP側の情報でIdP側のどのユーザの情報かを特定することはできない。

- ☑ persistent-id による連携

SP側でユーザプロファイルを保存する場合、IdP側から毎回同じIDを受け渡す必要がある。ShibbolethではeduPersonTargetedIDを利用することで、元のIDを特定できない形で固定IDを受け渡すことができる。Computed ID Data Connectorを用いる方法と、 Stored ID Data Connectorを用いる方法を調査した。

まとめ

Shibboleth SP の実装を用いて軽量アプリケーションに対するアクセス制御方法を調査した結果、ユーザの利便性を保つためには簡単な Wiki アプリケーションにおいても注意が必要であることが確認できた。また、Ruby on Rails のような Web アプリケーションフレームワークにおけるユーザ認証の実装について調査することで、MVC モデルにおけるユーザ認証、アクセス制御の実現方法の基礎について理解することができた。今後の課題として、具体的に運用されるアプリケーションへのShibbolethの導入や、より簡易に認証、アクセス制御機能が導入できるように、restful-authentication のような plugin として SSO 対応の認証機能を提供していくことなどが考えられる。

本報告で述べた内容の詳細は以下のWikiページにまとめられている。

Wikiページ（http://ssowiki.nii.ac.jp/）「00_各参加機関の取り組み＞京都産業大学」