仮想環境によるOpenAMを用いた認証基盤の紹介

An introduction of an authentication system using OpenAM in a virtual server environment.

花阪 元伸* 山崎 玲*
Motonobu Hanasaka, Akira Yamazaki

本稿では、仮想化やクラウド化のWebAPのサーバ構築への影響と、シングルサインオン（以下SSOとする）によるユーザ・管理者への利点ついて簡単に触れ、SSOシステムの一つである、OpenAMによるSSO構成について幾つかのパターンを紹介する。

We describe the impact on WebAP server virtualization and cloud computing, the benefits to users and administrators in Single Sign-On System（SSO）. We introduce some of the pattern for the SSO configuration using the OpenAM.

1. まえがき

一般に、Webアプリケーション（以下WebAP）やコンピュータ自体など、何らかのコンピュータ資源を誰かに利用させるにあたり、次の事項に対応することを求められる。

・情報へのアクセスを認められた者だけが、その情報にアクセスできる状態を確保すること。(1)

このために、コンピュータ資源等に少なくとも、次の２つの機能が必要となる。

表１　アクセス制限のための機能

機能	内容
認証機能	アクセスを行っている者を識別し確認する機能
認可機能	ある者が、そのコンピュータ資源上で何ができて何ができないかを決定する機能

これらの機能を実装する機構を、本稿では、認証認可機構と呼ぶ。

仮想化環境の利用により、気軽にWebAP毎にサーバを立てることが可能となった。しかし、一人のユーザが多数のWebAPを利用する時に、それぞれ個別にログインして回ることは不便である。また、個々のWebAPにおいて、ユーザ名やパスワードと共にユーザの情報を一々登録し、管理ポリシに応じて維持管理することは、管理者の作業負荷等が大変大きい。

本稿ではこの問題を解決する方法として、複数のWebAPに共通して導入できる、基盤としてのシングルサインオンシステム（以下SSOシステム）の紹介を行う。

SSOシステムとは、一度ログインすると、複数のWebAP等をシームレスに利用できるようにするシステムとなる。今回紹介するSSOシステムは、ForgeRock社（http://forgerock.com）が開発するOSSのOpenAMである。

本稿に関しては、上記SSOシステムだけではなく、一般のWebAP向けSSOシステムに対して概ね当てはまるのではないかと考える。

2. 仮想化と認証認可機構

ユーザ管理機能を除いて、WebAP等で専用の認証認可機構を実装することは難しいことではない。また、特定の条件において特定のユーザのみアクセスの許される情報がある場合等の細かなアクセス制御が必要な場合には、専用の認可機能の実装が必須となる。

WebAP等で専用の認証認可機構を持たず、ユーザ情報やアクセス条件を、共用のユーザ情報・認証認可サーバに保管し、各々のWebAPサーバからこれらの情報を参照する場合は、WebAP等とユーザ情報・認証認可サーバの間で、下記の内容等について、色々なすり合わせが必要となる。

・WebAPとユーザ情報・認証認可サーバとの間の通信規約
・ユーザ情報・認証認可サーバで保管する必要がある情報の種類
・情報の利用条件等

特にWebAPサーバとユーザ情報・認証認可サーバで管理組織等が異なる場合、上記すり合わせ自体にコストがかかる場合がある。

仮想化環境が広まる前までは、物理サーバ上に直接Web

＊つくば事業部 第六技術部

サーバを配置する必要があった。WebAP毎にサーバを構築することは物理サーバを購入することと同義である。このコストを低減するために、既設のサーバにWebAPのみを追加配置することも多く、追加配置作業の一環として、上記のすり合わせを実施することが多かった。

しかし、仮想サーバ環境やクラウド環境の場合、WebAP毎にサーバを構築したとしても、物理的にサーバ等を購入する必要がないため、あまりコストがかからない。これにより、面倒なすり合わせを避け、WebAP内にユーザ情報及び認証認可機能等をすべて取り込み、自己完結することができるようになった。

上記のようにWebAPが自己完結した場合、ユーザには利用するWebAP毎に個別にパスワードやユーザ情報等を登録し、これを維持管理する必要が発生する。これは、結果的に、情報システム全体として見た場合、覚えにくい多数のIDとパスワードを、機密性を高く保持するという難易度の高い作業を、一般のユーザに要求することになる。この要求は、一般のユーザには無理な要求となるため、複数のサービスで同じパスワードを使いまわす等、何らかのセキュリティ上のリスクとなることが考えられる。[(2)]

また、Internet向けのWebAP等の場合、FacebookやGoogle等のソーシャル系の認証が使えず、WebAP独自のユーザ登録が必要である場合、それだけで、ユーザに避けられる要因となる。

管理の面から見て、ユーザの情報を個々のWebAPで独立に維持管理することは、全体として無駄なコストとなる。特にエンタープライズシステムにおいて、定期的なパスワード変更が強制されたり、人事異動等によるユーザの属性情報の変更が多数発生した場合、管理者及びユーザが、各々のWebAPで個別に関係する作業を行うことになるため、大きなコストとなる。

上記の問題は、SSOシステムを導入することで解決することができる。

ユーザは統一されたログイン画面からログインすることで、SSOがサポートする全てのWebAPをシームレスに使用することができ、利便性が高まる。

管理側から見たメリットとしては、下記のものがある。

・WebAP側の認可の条件が非常に単純な場合、SSO側で認可を全て行うことができ、WebAPを単純にできる。
・WebAPで共通に利用するユーザに関する情報を集中管理することで、ユーザ管理のコストを抑えることができる。
・WebAPを作成する者のスキルレベルによらず、安定したセキュリティレベルを維持できる。
・ユーザがいつどのWebAPを利用したかの監査が容易となる。

一方、WebAP側では次の何れかの作業が必要となる。

・SSOに対応するように改修する。
・WebAPには手を入れずに、SSOとWebAPを接続するための糊づけに当たる部分を作成する。
・新規開発時に対象となるSSOに対応したWebAPを開発する。

3. SSOの方式

SSOを実現するための構成には次の２種類があげられる。

1. WebAP部分でSSOに対応するもの。（Agentタイプ）
2. WebAPより前にリバースプロキシをおいてSSOに対応するもの。（リバースプロキシタイプ）

それぞれについて、図示する。

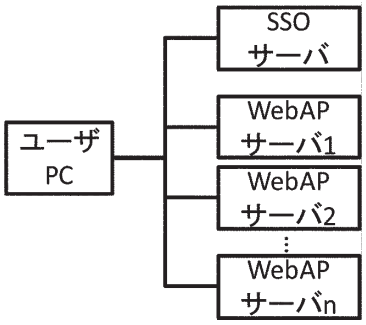

図１　AgentタイプのSSO

AgentタイプのSSO　について、必要な機構は次の通り

・SSOを行うSSOサーバ
・WebAP上でアクセス制限を実施する部分。

ユーザからWebAPへのアクセスでは次のように処理が行われる。

1. SSOサーバに対して、どのユーザがログインしてアクセスしているのか確認する
2. SSOサーバに対して、ログインユーザがアクセス先にアクセスしても良いかを確認する
3. 問題がなければ、WebAPへのアクセスが行われる。

これにより、ユーザはSSOサーバにログインしていれば、各々のWebAPでユーザ認証をする必要が無くなる。

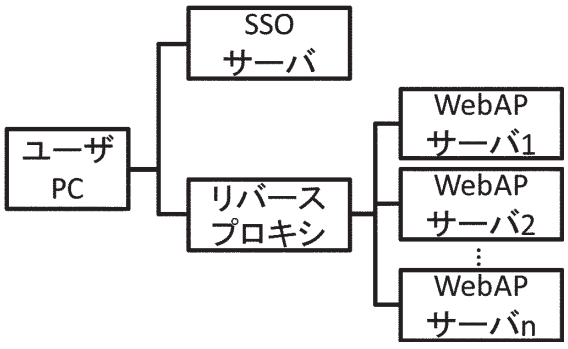

図２　リバースプロキシタイプのSSO

リバースプロキシタイプのSSOについて、必要な機構は次の通り

・SSOを行うSSOサーバ

・通信変換及びアクセス制限を実施するリバースプロキシ

なお、この構成では、SSOサーバとリバースプロキシが一体になっているものも多い。

ユーザからリバースプロキシにアクセスがあった際に、リバースプロキシ上で次のような処理が行われる。

1. SSOサーバに対して、どのユーザがログインしてアクセスしているのか確認する。
2. SSOサーバに対して、ログインユーザがアクセス先にアクセスしても良いかを確認する。
3. ユーザがWebAPにログインしていない場合、WebAPに対して、ユーザ毎に予め登録されたユーザIDと認証情報を送信してWebAPにログインする。
4. リバースプロキシサーバへのアクセスを変換して、WebAPにアクセスする。

これにより、ユーザは一度SSOサーバにログインしていれば、其々のWebAPで個別に認証を行う必要が無くなる。

AgentタイプのSSOとリバースプロキシタイプのSSOの構成を比べると、Agentタイプは構成及び動作がシンプルという利点があり、リバースプロキシタイプはWebAP自体がSSOを想定していないものでも対応できるという利点がある。

OpenAMは関連ソフトウエアと合わせて利用することにより上記どちらの方法にも対応する。

今回紹介するOpenAM及び関連ソフトウエアは下記の通りである。

表２　OpenAM及び関連ソフトウエア

機能	名称	備考
SSO サーバ	OpenAM	SSO 機能本体
ポリシエージェント	OpenAM PolicyAgent	SSO アクセス制限施行部
LDAP サーバ	OpenDJ	ユーザ情報を保管
リバースプロキシサーバ	OpenIG	
IDM サーバ	OpenIDM	複数の情報源からのユーザ情報の取得と連携および、ユーザ管理用サービスの提供

IDMサーバについては、これまで紹介されていないが、「5.OpenIDMによるユーザ情報管理を行う構成」で紹介する。

以降では、エンタープライズ向けのSSOシステムとして、OpenAMを採用した構成の例をいくつかあげる。

なお、本稿では、インターネット向けSSOシステムとして、OpenAMを構成した場合の例は含まない。

4.　アクセス制限に関する構成

本項では、「3. SSOの方式」のアクセス制限の方法に着目した構成を紹介する。

実際の構成では、個々のWebAPの用件に合わせた結果、以下に述べる２つの方法を組み合わせて利用することもある。

4.1　WebAP部分でSSOに対応するもの

本構成は、下記の構成となる。

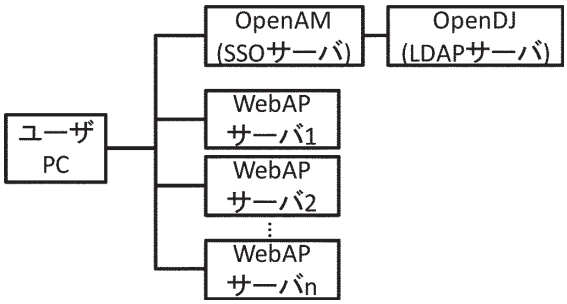

図３　OpenAMによるAgentタイプSSO

この構成はOpenAMを用いたSSOシステムでもかなり単純な例である。WebAPの改修ができるのであれば、多くの場合、小規模な改修で対応可能だと思われる。OpenAMを利用した場合、アクセス制限自体を施行するのは各Webサーバ等に配置されるOpenAM PolicyAgentで、これがWebサーバ等に組み込まれてアクセス制限を実施する。

OpenAM自体は、認証と、OpenAM PolicyAgentからの要求による認可の判断を実施する。

この構成の特徴は、下記の通りとなる。

・WebAPを提供するWebサーバ自体にOpenAM PolicyAgentを配置しアクセス制限を実施する。

・ユーザ情報をLDAPサーバであるOpenDJに保管して、OpenAMはこれを参照して動作する。

この構成のメリット及びデメリットは下記の通りとなる。

メリット

・ユーザに対して、SSOが提供できる。（ユーザはOpenAMに対してログインするだけで良く、WebAP1,WebAP2に個々にログインする必要がない。）

・構成がシンプルで分かりやすい。
・ログインユーザの識別方法やユーザ情報の入手に、OpenAMやLDAPが利用できる。
・ユーザ情報がLDAPに集中しているため、統一されたユーザ管理機能を開発することで、個々のWebAPでユーザ管理の機能が必要なくなり全体的に見て開発工数が低減できる。

デメリット
・OpenAM PolicyAgentをWebサーバに導入して、アクセス制限を行うため、OpenAM PolicyAgentに対応したWebサーバしか対応できない。
・WebAP等によるログインユーザの識別方法等が、OpenAMで提供する幾つかの方法に合致する必要がある。
・ユーザ情報の入手先が、OpenAMかLDAPサーバに限定される。
・上記に合致していない場合、WebAPの改修が必要となる。

4.2 WebAPより前にリバースプロキシをおいてSSOに対応するもの

本構成は、下記の通りとなる。

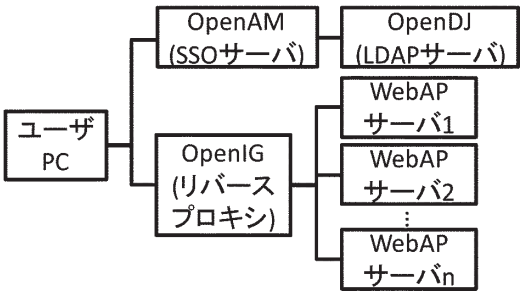

図４　OpenAMによるリバースプロキシタイプSSO

この構成はWebAPが、SAML等他の認証機構と連携する機能を有せず、ブラックボックスとなっており改修などが不可能な場合に利用される構成となる。

この構成の特徴は、下記の通り。
・コンテンツを提供するWebサーバの前に、リバースプロキシであるOpenIGを構成する。
・OpenIGにOpen AM PolicyAgentを配置しアクセス制限を実施する。
・OpenIGから、Webサーバ上のWebAPに対する代理認証等を実施する。

代理認証とは、SSO認証済みユーザからのアクセスに関して、ユーザの替わりに、OpenIGがWebAPに代理でログイン処理（ユーザID,Password）の入力等を実施することである。この代理認証の処理は、ユーザから見えないところで実施されるため、ユーザはあたかもWebAPに対して追加のログイン処理をせずに、シームレスにアクセスしているように見える。

この構成のメリット及びデメリットは下記の通りとなる。

メリット
・WebAPの改修をせずに、ユーザはSSOの恩恵を受けられる。

デメリット
・OpenIGをリバースプロキシとして立てる必要がある。
・OpenIGにリバースプロキシや代理認証の複雑な設定が必要となる。
・WebAPでユーザ管理や、OpenIGによる代理ログイン処理用のパスワード等の管理を管理者等が実施する必要がある。

なお、OpenIGの設定は、JSON形式のファイルを用いて実施する。

正しく構成するためには、複雑な設定が必要なため、構成を良く検討し、十分テストを行う必要がある。低コストで構築運用するためには、セキュリティ要件を検討したうえで、割り切った設計が求められる。

（代理認証用パスワードは長期にわたり利用する。WebAPへの通信路をOpenIGに固定してWebAP側では認証を行わないなど。）

5. OpenIDMによるユーザ情報管理を行う構成

ここでは、SSOシステムにOpenIDMを配置し、ユーザ情報に関する管理機能を強化した構成について紹介する。本構成は、「4. アクセス制限に関する構成」のどの構成に対しても追加できる。

本構成をとることにより、ユーザID・パスワードを含めたユーザ情報について、SSOシステム内外での連携を強化した基盤が構築できる。

本構成は、下記の通り。

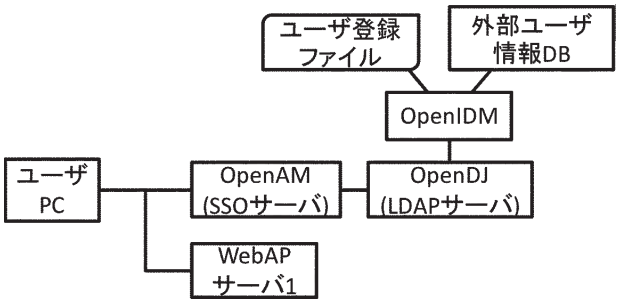

図５　OpenIDMを加えた構成

この構成は下記のような場合に用いられる。
・外部にユーザ情報（または人情報）管理システムがあ

り、この情報を取り込んでSSOのユーザ情報を作成する場合。
・複数のユーザ情報レポジトリとSSOのユーザ情報を連携し、一貫性のある状態を維持する必要がある場合。
・SSOシステムの外に対して、Webサービスやワークフローの機能としてユーザ管理機能を提供する場合。

OpenIDMは、内部的にSQLDBによるレポジトリを持ち、SSOのユーザ情報等の管理を行うサービスである。

以下の機能をもつ。
・他のユーザ情報源と、OpenDJ間のPushPull型の情報連携機能。
・ユーザ情報をJSON形式で提供し、同様に取り込み可能なREST APIによるユーザ管理用Webサービス。
・BPMNによるワークフローの提供。
・簡単な管理画面の提供。

情報連携機能は、ユーザ情報源としてファイル、SQLDB、LDAP等が利用できる。OpenAMで認証用ユーザ情報源として参照するOpenDJもLDAPの情報源として扱われる。

OpenIDMは、定期的に各情報源にアクセスし、更新された情報を内部レポジトリに反映する。これにより、内部レポジトリが更新された場合、各情報源を更新された情報で更新する。また、REST API等で直接内部レポジトリが変更された場合は、即時に各情報源を更新する機能を持つ。

これらの機能の組み合わせで、OpenIDMは全ての情報源のデータの整合性を維持する。

この連携機能は、先のOpenIGでの代理認証用に必要な、各WebAPでのユーザ管理（及びパスワードの維持）に関して支援を提供する。

なお、OpenIDMの設定は、JSON形式のファイルを用いて実施する。各種情報源と内部レポジトリとの間の情報のコンバートの際には、JavaScriptによるスクリプティングを実施することができる。

6. PKIを利用する構成

ここでは、OpenAMの豊富な認証機能の例として、ユーザIDとパスワードの入力ではなく、ユーザが所有するクライアント証明書を用いたSSOの例を示す。

本構成は、下記の構成となる。

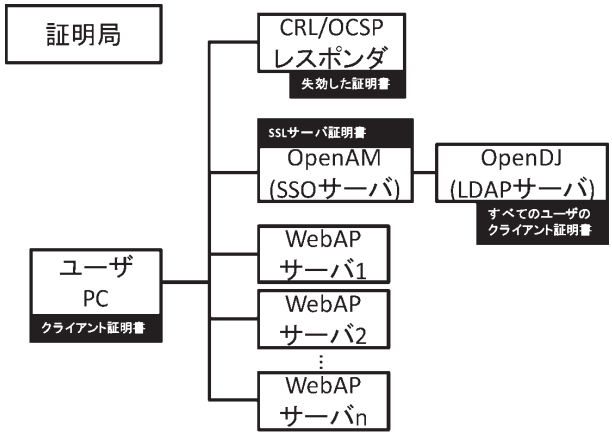

図６　クライアント証明書による認証構成

この構成ではユーザにPKIの鍵と証明書（以下クライアント証明書）を配り、これによる認証を実施するものである。このため、下記の構成を追加する。
・クライアント証明書を発行する証明局の構成または、外部の証明書発行サービスの利用
・クライアント証明書の照合を行うために、OpenDJのユーザ情報にクライアント証明書の保管（オプション）
・クライアント証明書のバリデーションのために、CRL、OCSPレスポンダへの準備（オプション）

この構成のメリット及びデメリットは下記の通りとなる。

メリット
・ユーザが明示的にパスワードを提示することなく、クライアント証明書を用いてシームレスに認証することができる。

デメリット
・クライアント証明書の作成・配布などの管理作業が発生する。
・ユーザがクライアント証明書を利用するための何らかの手続きが必要。
・ICカードによるPKI等を運用するためには、加えてハードウエア等の設備投資が必要。

7. むすび

本稿では、OpenAMによる、エンタープライズシステムで用いられそうなSSOの構成を紹介した。より進んだ構成としては、Windows Desktop SSOの機能を用いて、WindowsのログオンによりSSOを開始するなどの構成がある。これが適切に構成できた場合、ユーザは、Windowsにログインするだけで、全てのエンタープライズアプリケーションをシームレスに利用可能となる。

OpenAMでは、本稿には含まれないソーシャル系IDを用いた、クラウド連携の機能や、Internet向けのSSO連携の機能も含まれている。将来的には、社内システムやBtoBシステムにおいてもソーシャル系IDを用いたSSOが要求されるのではないだろうか。

そのためには、認証システムに対するより深い知見が必要であると考える。

参考文献

（1）情報セキュリティ（http://ja.wikipedia.org/wiki/情報セキュリティ），ウィキペディア（2014）

（2）「覚えられない」を前提にしたパスワード管理術とは？（http://www.atmarkit.co.jp/ait/articles/1311/08/news128.html），高橋睦美, @IT（2013）

（3）ForgeRock社WebSite（http://www.forgerock.com/），ForgeRock社（2014）

（4）ForgeRock Documentation（http://docs.forgerock.org/en/index.html），ForgeRock社（2014）

執筆者紹介

花阪　元伸

1996年入社。つくば事業部第六技術部第三グループ所属。基幹サーバ・大型計算機 及び オープンソースを使用したSI・運用支援業務に従事。

山崎　玲

1997年入社。つくば事業部第六技術部第一グループ所属。基幹サーバ・ネットワークシステム 及び オープンソースを使用したSI業務に従事。