Tomcat Policy Agent

リファレンスマニュアル

オープンソース・ソリューション・テクノロジ(株)

作成日: 2012 年 9 月 7 日

更新日: 2012 年 9 月 7 日

リビジョン: 1.0

目次

1. 要旨

本文書は Tomat Policy Agent の利用マニュアルです。

1.1 概要

- Tomcat Policy Agent のインストール方法や設定方法を説明します。

1.2 Agent インストール要件

Tomcat Policy Agent には次の要件があります。

- Apache Tomcat 6.0x(※)
- JDK 1.6.x 以降

※ 下記以外の Tomcat 環境では正常にインストールできない場合があります。インストール時にエラーとなる、またはインストール後に Tomcat が正常に起動しない際はお問い合わせ下さい。

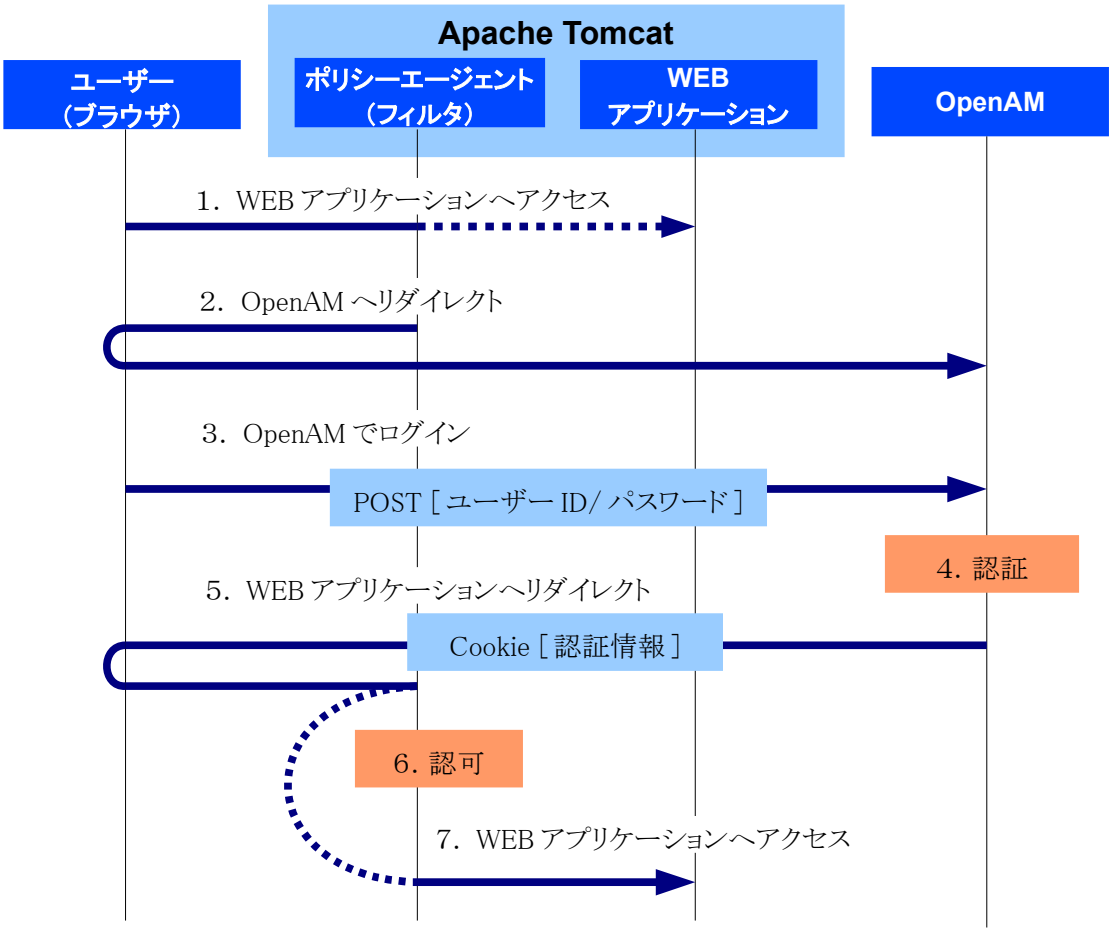
- Apache ダウンロードサイトから入手した Tomcat
- Red Hat 又は互換 OS 提供の RPM パッケージ版 Tomcat

2. ポリシーエージェント概要

ポリシーエージェントは保護された WEB リソースへアクセスするユーザーのリクエストを止め、OpenAM による認証・認可が済むまでリソースへのアクセスを拒否します。

2.1 処理概要

「保護対象の WEB アプリケーション」-「ポリシーエージェント」-「OpenAM」によるシングルサインオンのフローを説明します。



1. ユーザーは保護対象の WEB アプリケーションにアクセスします。
2. ポリシーエージェントは WEB アプリケーションへのアクセスを止め、OpenAM のセッション Cookie を利用して認証済みかどうかを判断します。認証済みでない場合は OpenAM へリダイレクトします
3. ユーザーは OpenAM でログインを行います。

4. OpenAM は POST されたユーザー ID とパスワードで認証を行います。

5. 認証が終了したら、OpenAM はセッション Cookie を付加して WEB アプリケーションへリダイレクトします。ポリシーエージェントはリダイレクトされた WEB アプリケーションへのアクセスを再度止めて、OpenAM のセッション Cookie から認証済みかどうかを判断します。

6. ポリシーエージェントは WEB アプリケーションへのアクセスを許可します。

7. ユーザーは WEB アプリケーションへアクセスします。

2.2 認証・認可

ポリシーエージェントは OpenAM と連携して認証と認可を行います。認証と認可の違いは以下の通りです。

- 認証 ― ユーザーが本人であることを確認する
- 認可 ― 認証されたユーザーが適切なリソースのみにアクセスするように権限を制御する

2.2.1 認証

ポリシーエージェントはユーザーが認証されていない場合に OpenAM にリダイレクトして認証を促します。

ただし、例外として認証を必要としない URL を OpenAM で設定することができます。設定方法は『3.2.1 適用されない URL の設定』を参照してください。

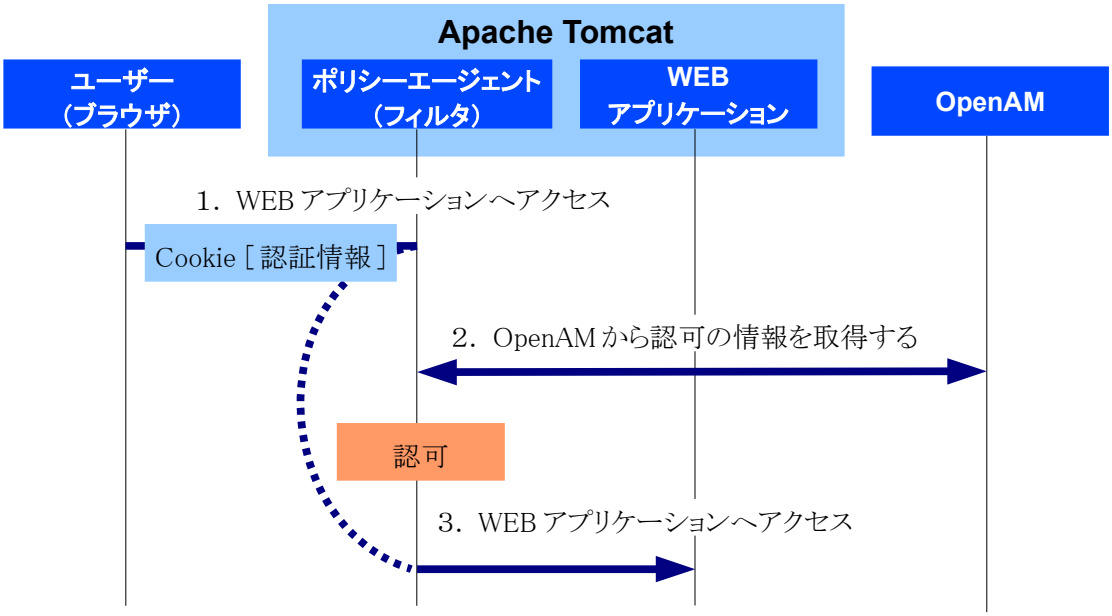
2.2.2 認可

認可は以下の方法があります。

- OpenAM で設定するポリシーによりポリシーエージェントが認可を行う方法
- ポリシーエージェントから Cookie やヘッダーで渡されるユーザーの属性を利用して保護対象の WEB アプリケーション自身が判定する方法
- Tomcat の認証フレームワークによって認可を行う方法

なお、ここではユーザーは既に OpenAM での認証を終えているものとし、認証のフローは省略します。

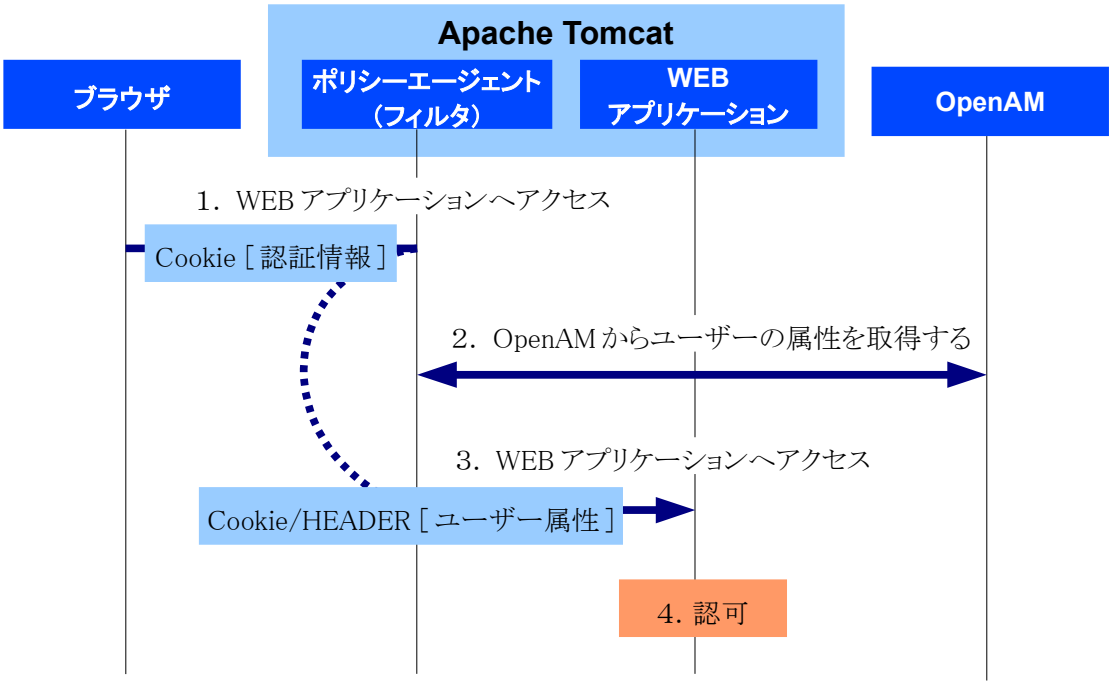
ポリシーエージェントによる認可



1. ユーザーは保護対象の WEB アプリケーションへアクセスします。
2. ポリシーエージェントは WEB アプリケーションへアクセスを止め、OpenAM のセッション Cookie から認証済みかどうかを判断します。認証済みの場合は OpenAM と通信して認可を行い、WEB アプリケーションへのアクセスを許可します。
3. ユーザーは WEB アプリケーションへアクセスします。

なお、ポリシーエージェントで認可を行う場合は OpenAM でポリシーを設定する必要があります。設定方法は『5 ポリシーの設定』を参照してください。

保護対象の WEB アプリケーションによる認可

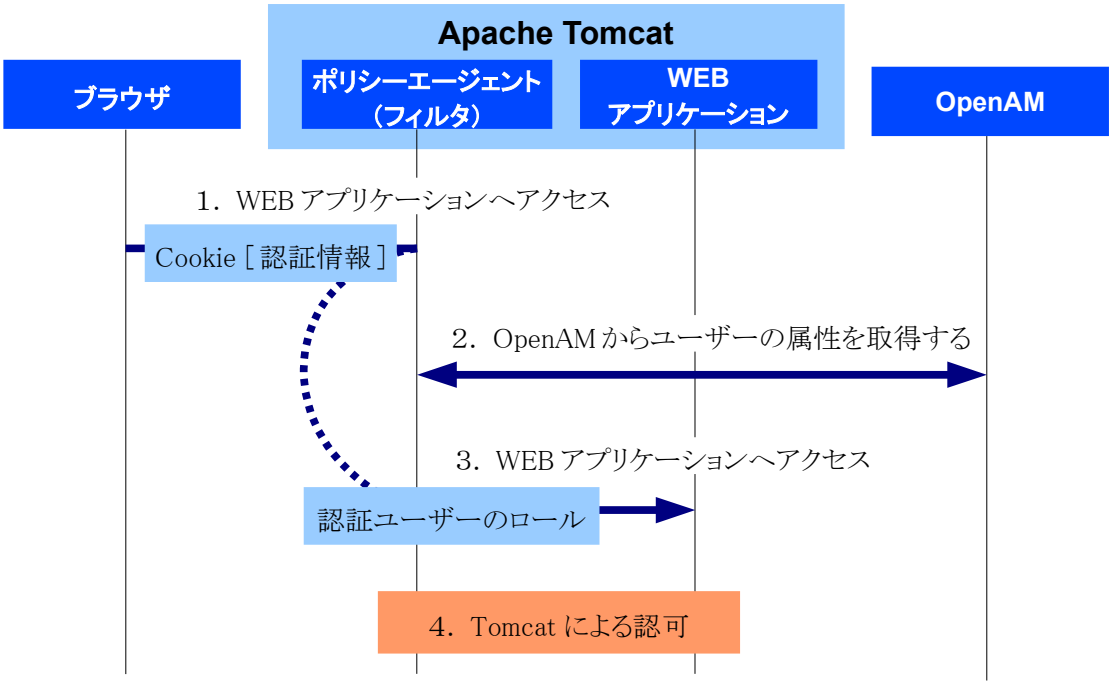


1. ユーザーは保護対象の WEB アプリケーションへアクセスします。
2. ポリシーエージェントは WEB アプリケーションへアクセスを止め、OpenAM のセッション Cookie から認証済みかどうかを判断します。認証済みの場合は OpenAM と通信してユーザー情報を取得します。
3. ポリシーエージェントは WEB アプリケーションへのリクエストに Cookie または HEADER を利用してユーザー情報を付加します。ユーザー属性は OpenAM のデータストア(LDAP など)から取得したものです。
4. WEB アプリケーションはリクエストに付加されたユーザー情報を利用してアクセス許可・拒否を判断します。

   エージェントが付与するヘッダ、Cookie により認可するロジックが無い場合は追加開発する必要があります。

なお、WEB アプリケーションが認可を行う場合はポリシーエージェントで認可を行わないための設定が必要です。『5 ポリシーの設定』ですべてのリソースのアクセスを許可してください。またリクエストに付加するユーザー属性の設定方法は『3.2.2 ユーザー属性の取得の設定』を参照してください。

Tomcat の認証フレームワークによる認可



1. ユーザーは保護対象の WEB アプリケーションへアクセスします。
2. ポリシーエージェントは WEB アプリケーションへアクセスを止め、OpenAM のセッション Cookie から認証済みかどうかを判断します。認証済みの場合は OpenAM と通信してユーザー情報を取得します。
3. ポリシーエージェントは Tomcat の認証フレームワークによる FORM 認証を自動で行い、Tomcat に対して認証ユーザーのロールを返却します。
4. Tomcat は認証ユーザーのロールに従ってアクセスを許可します。

なお、Tomcat が認可を行う場合はポリシーエージェントで認可を行わないための設定が必要です。『5 ポリシーの設定』ですべてのリソースのアクセスを許可してください。設定方法については『6 Tomcat の認証フレームワークを利用した認可』を参照してください。

3. ポリシーエージェントの設定

この章では OpenAM でポリシーエージェントの設定を行う方法を説明します。『3.1 基本設定』は『4 ポリシーエージェントのインストール』の前に実施してください。『3.2 動作設定』はエージェントの動作に対する必要性に応じて実施してください。

3.1 基本設定

3.1.1 エージェントプロファイルの作成

OpenAM 側でポリシーエージェント設定(エージェントプロファイル)を作成します。

1. OpenAM に管理者ユーザーでログインします。
2. 「アクセス制御」→「レルム名」→「エージェント」→「J2EE」を選択します。
3. 「エージェント」の「新規」をクリックします。
4. 各項目を入力します。
   a) 「名前」：エージェントプロファイルの名前を入力します(例として、“tomcat6”と入力したものとします)。
   b) 「パスワード」：エージェントの認証を行なうためのパスワードを入力します。ここで入力したパスワードは、エージェントをインストールする際に入力する必要があります。
   c) 「設定」：「集中」を選択します。
   d) 「サーバー URL」：OpenAM の URL を入力します。ポート番号も指定してください。
   e) 「エージェント URL」：エージェントの URL を入力します。ポート番号も指定してください。
5. 「作成」をクリックします。

3.2 動作設定

3.2.1 適用されない URL の設定

認証を必要としない URL を設定する方法を説明します。

1. OpenAM 管理コンソールの「アクセス制御」 → 「レルム」の対象レルム → 「エージェント」 → 「J2EE」 → 対象のエージェントを開きます。
2. 「アプリケーション」タブの「適用されない URI」にエージェントで保護しない URL を設定します。
3. 画面右上の「保存」をクリックして設定を保存します。

以上で完了です。

もし、保護するURLが限られている場合は、2の「適用されないURI」に保護するULRを設定し、「適用されないURLの反転」にチェックを入れて保存します。

3.2.2 ユーザー属性の取得の設定

HTTPヘッダーにユーザーデータストア（LDAPなど）のユーザー属性を格納し、Agentの保護下にあるアプリケーションにユーザー情報を渡して認可に利用するための設定方法について説明します。

1. OpenAM管理コンソールの「アクセス制御」→「レルム」の対象レルム→「エージェント」→「J2EE」を開きます。
2. 「エージェント」の一覧から、対象のエージェントをクリックし、「アプリケーション」タブを開きます。
3. 「プロファイル属性処理」の「プロファイル属性フェッチモード」から「HTTP_HEADER」を選択します。
4. 「プロファイル属性マップ」に以下の要領でマッピングを追加します。
    - 「マップキー」：ユーザーデーターストアの属性名（例：uid）
    - 「対応するマップ値」：ヘッダー名（例：UID）
5. 画面右上の「保存」をクリックし設定を保存します。

以上で完了です。

なお、情報をCookieに格納する場合は、3の「プロファイル属性フェッチモード」に「HTTP_COOKIE」を指定します。

4. ポリシーエージェントのインストール

この章ではWEBアプリケーションサーバーにポリシーエージェントをインストールする方法を説明します。なお、この章ではRed Hat Enterprise Linux 6 64bit版パッケージをインストールするものとします。他のOSのパッケージでも手順は同様です。

4.1 パッケージのインストール

rpmコマンドを使用して、RPMパッケージをインストールします。(「X」の部分にはビルドバージョンが入ります。)

```
# rpm -ivh osstech-base-3.0-X.el6.noarch.rpm ¥
osstech-openam-agent-tomcat6-3.0.3-X.el6.noarch.rpm
```

「¥(バックスラッシュ)」はコマンドラインの途中で改行を行うために入れています。「¥(バックスラッシュ)」を入れずに、全てのRPMファイルを一行で指定することも可能です。

4.2 エージェントのインストール

Tomcat Policy Agentのインストール手順を説明します。

なお、”Apacheダウンロードサイトから入手したTomcat”と”Red Hat又はOS提供のRPMパッケージ版Tomcat”では構成が異なるため、インストール手順が異なります。

”Apacheダウンロードサイトから入手したTomcat”にインストールする場合は『4.2.1 Apacheダウンロードサイトから入手したTomcatにインストール』を、”Red Hat又はOS提供のRPMパッケージ版Tomcat”にインストールする場合は『4.2.2 Red Hat 又は互換OS提供のRPMパッケージ版Tomcatにインストール』を参照してください。

4.2.1 Apacheダウンロードサイトから入手したTomcatにインストール

Apacheサイトのダウンロード版Tomcat環境にエージェントをインストールする手順を示します。なお、Tomcatは”/usr/share/tomcat6”に配置し、起動および停止は”tomcat”ユーザーにより付属のスクリプトで行うものとします。

1. 事前にインストールに必要な情報を収集しておきます。

   『3.1.1 エージェントプロファイルの作成』の手順で設定されたOpenAM側の情報を確認してください。

   - エージェントプロファイルの名前(例:“tomcat6”)
   - パスワード(例:“password”)
   - サーバーURL(例:“http://sso.example.co.jp:8080/openam”)
   - エージェントURL(例:“http://agent.example.co.jp:8080/agentapp”)

2. Tomactを停止します。

```
# sudo -u tomcat /usr/share/tomcat6/bin/shutdown.sh
```

3. パスワードファイルを作成します。

エージェントが OpenAM と通信する際に使用するパスワードをファイルに記述します。パスワードファイルはエージェントインストール時にのみ必要であり、インストール後は削除してもかまいません。パスワードファイルにはパスワードを平文で記述します。パスワードファイルは一行で記述し、改行コードを含まないものである必要があります。echo コマンドの”-n”オプションや tr コマンドを使用してパスワードファイルを生成します。

```
# echo -n  “password”  > passwd
```

もしくは

```
# vim passwd
(パスワードを記述して保存)
# tr -d '¥n' < passwd
```

以下のようなファイルとなります(末尾に改行コードを含みません)。

```
password
```

以上でパスワードファイルの作成は完了です。記述したパスワードはエージェントインストール時に暗号化されてエージェントの設定ファイルに保存されます。

4. エージェントのインストーラーを実行します。

```
# cd /opt/osstech/share/openam-agent-tomcat6/bin
# ./agentadmin --install
Please read the following License Agreement carefully:

[Press <Enter> to continue...] or [Enter n To Finish]
n      #「n」を入力

Do you completely agree with all the terms and conditions of this License
Agreement (yes/no): [no]: yes     #「yes」と入力
```

Tomcat の設定ファイルが存在するディレクトリのパスを入力します。

```
Enter the complete path to the directory which is used by Tomcat Server to
store its configuration Files. This directory uniquely identifies the
Tomcat Server instance that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Tomcat Server Config Directory Path
[/opt/apache-tomcat-6.0.14/conf]: /usr/share/tomcat6/conf
```

OpenAM の URL を入力します。

```
Enter the URL where the OpenSSO server is running. Please include the
deployment URI also as shown below:
(http://opensso.sample.com:58080/opensso)
[ ? : Help, < : Back, ! : Exit ]
OpenSSO server URL: http://sso.example.co.jp:8080/openam
```

CATALINA_HOME のディレクトリを指定します。

```
$CATALINA_HOME environment variable is the root of the tomcat
installation.
[ ? : Help, < : Back, ! : Exit ]
Enter the $CATALINA_HOME environment variable: /usr/share/tomcat6
```

ポリシーエージェントを Tomcat 全体に適用する場合は”true”、配備するアプリケーション毎に設定する場合は”false”を入力します。

```
Choose yes to deploy the policy agent in the global web.xml file.
[ ? : Help, < : Back, ! : Exit ]
Install agent filter in global web.xml ? [true]: false
```

エージェントの URL を入力します。

```
Enter the Agent URL. Please include the deployment URI also as shown below:
(http://agent1.sample.com:1234/agentapp)
[ ? : Help, < : Back, ! : Exit ]
Agent URL: http://agent.example.co.jp:8080/agentapp
```

事前に OpenAM 側で作成しておいた、エージェントプロファイルのプロファイル名を入力します。

```
Enter the Agent profile name
[ ? : Help, < : Back, ! : Exit ]
Enter the Agent Profile name: tomcat6
```

エージェントのパスワードファイルを指定します。

```
Enter the path to a file that contains the password to be used for identifying
the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /opt/osstech/etc/openam-agent-tomcat6/passwd
```

これまで入力したパラメーターの確認メッセージが表示されます。問題がない場合は「1」を選択し Enter を入力します。

```
---------------------------------------------
SUMMARY OF YOUR RESPONSES
---------------------------------------------
Tomcat Server Config Directory : /usr/share/tomcat6/conf
OpenSSO server URL : http://sso.example.co.jp:8080/openam
$CATALINA_HOME environment variable : /usr/share/tomcat6
Tomcat global web.xml filter install : false
Agent URL : http://agent.example.co.jp:8080/agentapp
Agent Profile name : tomcat6
Agent Profile Password file name : /opt/osstech/etc/openam-agent-tomcat6/passwd

Verify your settings above and decide from the choices below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]:
```

インストールログが表示されます。

```
Updating the /usr/share/tomcat6/bin/setenv.sh script with the
Agent configuration JVM option ...DONE.
DONE.

Creating directory layout and configuring Agent file for Agent_001
instance ...DONE.

Reading data from file /opt/osstech/etc/openam-agent-tomcat6/passwd and encrypting it
...DONE.

Generating audit log file name ...DONE.

Creating tag swapped OpenSSOAgentBootstrap.properties file for instance
Agent_001 ...DONE.

Creating a backup for file /usr/share/tomcat6/conf/server.xml
...DONE.

Creating a backup for file /usr/share/tomcat6/conf/web.xml ...DONE.

Adding OpenSSO Tomcat Agent Realm to Server XML file :
/usr/share/tomcat6/conf/server.xml ...DONE.

Adding filter to Global deployment descriptor file :
/usr/share/tomcat6/conf/web.xml ...DONE.

Adding OpenSSO Tomcat Agent Filter and Form login authentication to
selected Web applications ...DONE.


SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/opt/osstech/share/openam-agent-tomcat6/Agent_001/config/OpenSSOAgentBootstrap.properties
Agent Configuration file location
/opt/osstech/share/openam-agent-tomcat6/Agent_001/config/OpenSSOAgentConfiguration.properties
Agent Audit directory location:
/opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/audit
Agent Debug directory location:
/opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/debug


Install log file location:
/opt/osstech/share/openam-agent-tomcat6/installer-logs/audit/install.log

Thank you for using OpenSSO Policy Agent
```

5. エージェントの設定フォルダに Tomcat 実行ユーザーのためのアクセス権を設定します。”logs”フォルダ配下には書き込み権限が必要です。

```
# cd /opt/osstech/share/openam-agent-tomcat6
# chown -R root:tomcat Agent_001
# chmod -R g+w Agent_001/logs
```

6. エージェントアプリケーションの WAR ファイルを Tomcat に配備します。

```
# cp /opt/osstech/share/openam-agent-tomcat6/etc/agentapp.war /usr/share/tomcat6/webapps/
```

7. 保護対象の WEB アプリケーションの web.xml を編集します。

   インストーラーの設定でポリシーエージェントを Tomcat 全体に適用していない場合、WEB アプリケーションの web.xml を編集する必要があります。<webapp>タグの下に次の設定を追加します。

```
<filter>
  <filter-name>Agent</filter-name>
  <display-name>Agent</display-name>
  <description>OpenAM Policy Agent Filter</description>
  <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>Agent</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>ERROR</dispatcher>
</filter-mapping>
```

8. Tomcat を起動します。

   Tomcat を正常に起動するためには、OpenAM で『3.1.1 エージェントプロファイルの作成』の操作が完了している必要があります。

```
# sudo -u tomcat /usr/share/tomcat6/bin/startup.sh
```

   正常に起動した場合、Tomcat に次のログが出力されます。

```
# tail -n 1 /usr/share/tomcat6/logs/catalina.out
INFO: Server startup in 810 ms
# tail -n 7 /opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/debug/debug.out
=====================================
Version: 3.0.3
Revision: 3.0.3-X
Build Date: 2012XXXX
Build Machine: build-X.lan.ostech.co.jp
=====================================
```

9. 必要に応じてパスワードファイルを削除します。

```
# rm /opt/osstech/etc/openam-agent-tomcat6/passwd
```

以上でポリシーエージェントのインストールは終了です。

4.2.2 Red Hat 又は互換 OS 提供の RPM パッケージ版 Tomcat にインストール

Apache サイトのダウンロード版 Tomcat 環境にエージェントをインストールする手順を示します。なお、Tomcat は”/usr/share/tomcat6”に配置し、起動および停止は付属のスクリプトで行うものとします。

1. 事前にインストールに必要な情報を収集しておきます。

   『3.1.1 エージェントプロファイルの作成』の手順で設定された OpenAM 側の情報を確認してください。

   - エージェントプロファイルの名前(例:“tomcat6”)
   - パスワード(例:“password”)
   - サーバー URL(例:“http://sso.example.co.jp:8080/openam”)
   - エージェント URL(例:“http://agent.example.co.jp:8080/agentapp”)

2. Tomact を停止します。

```
# /sbin/service tomcat6 stop
```

3. パスワードファイルを作成します。

   エージェントが OpenAM と通信する際に使用するパスワードをファイルに記述します。パスワードファイルはエージェントインストール時にのみ必要であり、インストール後は削除してもかまいません。パスワードファイルにはパスワードを平文で記述します。パスワードファイルは一行で記述し、改行コードを含まないものである必要があります。echo コマンドの”-n”オプションや tr コマンドを使用してパスワードファイルを生成します。

```
# echo -n  “password”  > passwd
```

   もしくは

```
# vim passwd
(パスワードを記述して保存)
# tr -d '¥n' < passwd
```

   以下のようなファイルとなります(末尾に改行コードを含みません)。

```
password
```

   以上でパスワードファイルの作成は完了です。記述したパスワードはエージェントインストール時に暗号化されてエージェントの設定ファイルに保存されます。

4. エージェントのインストーラーを実行します。

```
# cd /opt/osstech/share/openam-agent-tomcat6/bin
# ./agentadmin --install
Please read the following License Agreement carefully:

[Press <Enter> to continue...] or [Enter n To Finish]
n      #「n」を入力
```

```
Do you completely agree with all the terms and conditions of this License
Agreement (yes/no): [no]: yes    #「yes」と入力
```

Tomcat の設定ファイルが存在するディレクトリのパスを入力します。

```
Enter the complete path to the directory which is used by Tomcat Server to
store its configuration Files. This directory uniquely identifies the
Tomcat Server instance that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Tomcat Server Config Directory Path
[/opt/apache-tomcat-6.0.14/conf]: /usr/share/tomcat6/conf
```

OpenAM の URL を入力します。

```
Enter the URL where the OpenSSO server is running. Please include the
deployment URI also as shown below:
(http://opensso.sample.com:58080/opensso)
[ ? : Help, < : Back, ! : Exit ]
OpenSSO server URL: http://sso.example.co.jp:8080/openam
```

CATALINA_HOME のディレクトリを指定します。

```
$CATALINA_HOME environment variable is the root of the tomcat
installation.
[ ? : Help, < : Back, ! : Exit ]
Enter the $CATALINA_HOME environment variable: /usr/share/tomcat6
```

ポリシーエージェントを Tomcat 全体に適用する場合は”true”、配備するアプリケーション毎に設定する場合は”false”を入力します。

```
Choose yes to deploy the policy agent in the global web.xml file.
[ ? : Help, < : Back, ! : Exit ]
Install agent filter in global web.xml ? [true]: false
```

エージェントの URL を入力します。

```
Enter the Agent URL. Please include the deployment URI also as shown below:
(http://agent1.sample.com:1234/agentapp)
[ ? : Help, < : Back, ! : Exit ]
Agent URL: http://agent.example.co.jp:8080/agentapp
```

事前に OpenAM 側で作成しておいた、エージェントプロファイルのプロファイル名を入力します。

```
Enter the Agent profile name
[ ? : Help, < : Back, ! : Exit ]
Enter the Agent Profile name: tomcat6
```

エージェントのパスワードファイルを指定します。

```
Enter the path to a file that contains the password to be used for identifying
the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /opt/osstech/etc/openam-agent-tomcat6/passwd
```

これまで入力したパラメーターの確認メッセージが表示されます。問題がない場合は「1」を選択し Enter

を入力します。

```
------------------------------------------------
SUMMARY OF YOUR RESPONSES
------------------------------------------------
Tomcat Server Config Directory : /usr/share/tomcat6/conf
OpenSSO server URL : http://sso.example.co.jp:8080/openam
$CATALINA_HOME environment variable : /usr/share/tomcat6
Tomcat global web.xml filter install : false
Agent URL : http://agent.example.co.jp:8080/agentapp
Agent Profile name : tomcat6
Agent Profile Password file name : /opt/osstech/etc/openam-agent-tomcat6/passwd

Verify your settings above and decide from the choices below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]:
```

インストールログが表示されます。

```
Updating the /usr/share/tomcat6/bin/setenv.sh script with the
Agent configuration JVM option ...DONE.
DONE.

Creating directory layout and configuring Agent file for Agent_001
instance ...DONE.

Reading data from file /opt/osstech/etc/openam-agent-tomcat6/passwd and encrypting it
...DONE.

Generating audit log file name ...DONE.

Creating tag swapped OpenSSOAgentBootstrap.properties file for instance
Agent_001 ...DONE.

Creating a backup for file /usr/share/tomcat6/conf/server.xml
...DONE.

Creating a backup for file /usr/share/tomcat6/conf/web.xml ...DONE.

Adding OpenSSO Tomcat Agent Realm to Server XML file :
/usr/share/tomcat6/conf/server.xml ...DONE.

Adding filter to Global deployment descriptor file :
/usr/share/tomcat6/conf/web.xml ...DONE.

Adding OpenSSO Tomcat Agent Filter and Form login authentication to
selected Web applications ...DONE.


SUMMARY OF AGENT INSTALLATION
---------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
```

```
/opt/osstech/share/openam-agent-tomcat6/Agent_001/config/OpenSSOAgentBootstrap.properties
Agent Configuration file location
/opt/osstech/share/openam-agent-tomcat6/Agent_001/config/OpenSSOAgentConfiguration.properties
Agent Audit directory location:
/opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/audit
Agent Debug directory location:
/opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/debug


Install log file location:
/opt/osstech/share/openam-agent-tomcat6/installer-logs/audit/install.log

Thank you for using OpenSSO Policy Agent
```

5. Tomcat の起動パラメータ を設定します。

   4の操作終了後、Tomcat のインストールフォルダの bin 配下に Tomcat の起動パラメータを設定する setenv.sh が作成されます。**ただし、Red Hat 又は互換 OS 提供の RPM パッケージ版 Tomcat の場合は setenv.sh が読み込まれないため、手動で設定する必要があります。**

   まずは setenv.sh の内容を確認します。

```
# cat /usr/share/tomcat6/bin/setenv.sh
JAVA_OPTS="$JAVA_OPTS -Dopenam.agents.bootstrap.dir=/opt/osstech/share/openam-agent-tomcat6/Agent_001/config"
```

   tomcat6.conf に setenv.sh の内容を追記します。

```
# vi /usr/share/tomcat6/conf/tomcat6.conf
```

```
JAVA_OPTS="$JAVA_OPTS -Dopenam.agents.bootstrap.dir=/opt/osstech/share/openam-agent-tomcat6/Agent_001/config"
```

6. エージェントの設定フォルダに Tomcat 実行ユーザーのためのアクセス権を設定します。”logs”フォルダ配下には書き込み権限が必要です。

```
# cd /opt/osstech/share/openam-agent-tomcat6
# chown -R root:tomcat Agent_001
# chmod -R g+w Agent_001/logs
```

7. エージェントアプリケーションの WAR ファイルを Tomcat に配備します。

```
# cp /opt/osstech/share/openam-agent-tomcat6/etc/agentapp.war /usr/share/tomcat6/webapps/
```

8. 保護対象の WEB アプリケーションの web.xml を編集します。

   インストーラーの設定でポリシーエージェントを Tomcat 全体に適用していない場合、WEB アプリケーションの web.xml を編集する必要があります。<webapp>タグの下に次の設定を追加します。

```
<filter>
  <filter-name>Agent</filter-name>
  <display-name>Agent</display-name>
  <description>OpenAM Policy Agent Filter</description>
  <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
```

```
</filter>
<filter-mapping>
  <filter-name>Agent</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>ERROR</dispatcher>
</filter-mapping>
```

9. Tomcat を起動します。

   Tomcat を正常に起動するためには、OpenAM で『3.1.1 エージェントプロファイルの作成』の操作が完了している必要があります。

```
# /sbin/service tomcat6 start
```

   正常に起動した場合、Tomcat に次のログが出力されます。

```
# tail -n 1 /usr/share/tomcat6/logs/catalina.out
INFO: Server startup in 810 ms
# tail -n 7 /opt/osstech/share/openam-agent-tomcat6/Agent_001/logs/debug/debug.out
=====================================
Version: 3.0.3
Revision: 3.0.3-X
Build Date: 2012XXXX
Build Machine: build-X.lan.ostech.co.jp
=====================================
```

10. 必要に応じてパスワードファイルを削除します。

```
# rm /opt/osstech/etc/openam-agent-tomcat6/passwd
```

以上でポリシーエージェントのインストールは終了です。

5. ポリシーの設定

この章では、OpenAM エージェントのポリシー設定について説明します。

5.1 参照ポリシーの作成

※この操作はサブレルムを利用して構築した場合のみ必要です

サブレルムでポリシーを作成する場合は、事前に「/ (最上位のレルム)」で参照ポリシーを作成しておく必要があります。ここでは、参照ポリシーを新規作成する手順を説明します。

1. OpenAM 管理コンソールの「アクセス制御」→「/ (最上位のレルム)」→「ポリシー」を開きます。
2. 「新規参照」をクリックします。
3. 「一般」の「名前」を入力します。
4. 「ルール」の「新規」をクリックします。
5. 「サービスタイプ」は「URL ポリシーエージェント (リソース名あり)」を選択し、「次へ」をクリックします。
6. 「名前」にアクセス制御対象のリソースを表す名前を入力します。
7. 「リソース名」にアクセス制御対象のリソースのルートパスの URL を入力します。
8. 画面右上の「終了」をクリックします。
9. 「新規参照」画面に戻るため、「参照」の「新規」をクリックします。
10. 「値」から、このポリシーの参照を許可するレルムを選択します。「名前」に適当な名前を入力し、画面右上の「終了」をクリックします。
11. 「新規参照」画面に戻るため、画面右上の「了解」をクリックします。
12. 「ポリシーの編集」画面に戻るため、画面右上の「保存」をクリックし設定を保存します。ここで「保存」をクリックすることでポリシーが保存されます。

5.2 ポリシーの作成

ポリシーには主にルール・対象・条件の3つの要素があります。

ポリシーの作成手順としては、まずポリシー名を決定し、そのあとにルール・対象・条件を追加していきます。

ポリシー		
├	ルール	アクセス制御対象 URL に対する許可・拒否の設定
├	対象	アクセスを許可・拒否するユーザーの設定

└ 条件	アクセスを許可・拒否する条件の設定

5.2.1 ポリシー

まずはポリシー名を設定します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「ポリシー」を開きます。
2. 「新規ポリシー」をクリックします。
3. 「一般」の「名前」を入力します。
4. 画面右上の「了解」をクリックし設定を保存します。

ひきつづき、ルール・対象・条件を設定します。

5.2.2 ルール

アクセス制御対象の URL に対するアクセスの許可/拒否を設定します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「ポリシー」を開きます。
2. 編集するポリシーをクリックします。
3. 「ルール」の新規をクリックします。
4. 「サービスタイプ」は「URL ポリシーエージェント (リソース名あり)」を選択し、「次へ」をクリックします。
5. 「名前」にアクセス制御対象のリソースを表す名前を入力します。
6. 「リソース名」にアクセス制御対象のリソースの URL を入力します。この URL は「親リソース名」の配下に属するリソースを表す URL しか入力できません。また、URL にはワイルドカードを利用することができます。詳細は後述の『ワイルドカードについて』を参照してください。
7. 「アクション」でアクセス制御のルールを選択します。
8. 画面右上の「終了」をクリックします。**この状態では、ポリシーはまだ OpenAM に保存されていないため、ご注意ください。**
9. 「ポリシーの編集」画面に戻るため、画面右上の「保存」をクリックし設定を保存します。

ここで、「サービス iPlanetAMWebAgentService のリソース http://www.example.co.jp/ は現在の組織では管理できません。」いうエラーメッセージが表示された場合は、このポリシー設定の参照元である参照ポリシーで親リソースが定義されていません。「5.1 参照ポリシーの作成」の手順で参照ポリシーを定義してください。

注意点

OpenAM のポリシーでは、ユーザー(ブラウザ)から送出された HTTP リクエストのリソースパスとポリシーで定義されているリソースパスを厳格に比較します。たとえば、”http://www.example.co.jp/”などの URL にアクセスした場合に”http://www.example.co.jp/index.html”のコンテンツがユーザー(ブラウザ)に応答されることがありますが、

ポリシーのルールで”http://www.example.co.jp/”へのアクセスを許可しているだけでは、「403 Forbidden」エラーとなります。”http://www.example.co.jp/index.htm”というリソースへの明確なアクセス許可がルールに定義されていないためです。このため、アクセスを許可する場合は、そのリソースのパス(ファイルシステムのパスと一致)を正確にルールで定義してください。

ワイルドカードについて

「リソース名」にはワイルドカードを利用することができます。”http://www.example.co.jp/sampleapp/” の配下のすべてのコンテンツを許可する場合、”http://www.example.co.jp/sampleapp/*” と入力します。

ただし、”http://www.example.co.jp/sampleapp/content.jsp?param1=1”のようにURLにクエリストリングが付加されている場合は”http://www.example.co.jp/sampleapp/*?*”のように”?”をはさみ”*”を2回入力する必要があります。

5.2.3 対象

アクセスを許可する対象ユーザーを設定します。ここでは「OpenAM アイデンティティー対象」を対象として設定し、このレルムで管理されているユーザーにポリシーを適用する方法を説明します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「ポリシー」を開きます。
2. 編集するポリシーをクリックします。
3. 「対象」の新規をクリックします。
4. 「タイプ」で「OpenAM アイデンティティー対象」を選択します。
5. 「新規対象 - OpenAM アイデンティティー対象」画面で、以下の内容を入力します。
   a) 名前:この対象を識別する任意の名前です。
   b) 排他的:選択されていない場合はc)で設定するアイデンティティーに適用されます。選択されている場合は設定されていないアイデンティティーに適用されます。
   c) フィルタ:対象を検索するためのフィルタです。アイデンティティータイプから「ユーザー」もしくは「グループ」を選択し、「検索」をクリックすると検索結果が表示されます。検索結果からアクセス制御対象のユーザー/グループを選択し、「追加」/「すべてを追加」で「選択」エリアに追加します。
6. 画面右上の「終了」をクリックします。**この状態では、ポリシーはまだ OpenAM に保存されていないため、ご注意ください。**
7. 「ポリシーの編集」画面に戻るため、画面右上の「保存」をクリックし設定を保存します。

5.2.4 条件

アクセスを許可する条件を設定します。ここでは「LDAP フィルタ条件」を条件として設定し、LDAP フィルタで対象のユーザーエントリが見つかった場合にアクセスを許可する設定を説明します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「ポリシー」を開きます。

2. 編集するポリシーをクリックします。

3. 「条件」の新規をクリックします。

4. 「タイプ」で「LDAP フィルタ条件」を選択します。

5. 「新規条件- LDAP フィルタ条件」画面で、以下の内容を入力します。

   a) 名前:この条件を識別する任意の名前です。

   b) ldapFilter:LDAP で対象を検索するためのフィルタです。

6. 画面右上の「終了」をクリックします。**この状態では、ポリシーはまだ OpenAM に保存されていないため、ご注意ください。**

7. 「ポリシーの編集」画面に戻るため、画面右上の「保存」をクリックし設定を保存します。

5.3 ポリシーの削除

ポリシーを削除する手順は以下のようになります。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「ポリシー」を開きます。

2. 削除するポリシーを選択します。

3. 「削除」をクリックします。

6. Tomcat の認証フレームワークを利用した認可

この章では、Tomcat の認証フレームワークを利用した認可の設定について説明します。

6.1 OpenAM 側の設定

6.1.1 グループの設定

ここでは Tomcat のロールとなる OpenAM のグループを設定します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム →「対象」→「グループ」を開きます。
2. グループの「新規」をクリックして、表示された画面で ”ID” にグループ名を入力し、「了解」をクリックします。
3. 作成したグループをクリックして、表示された画面で「ユーザー」タブに切り替え、グループに参加させるユーザーを選択します。
4. 画面右上の「保存」をクリックして設定を保存します。

以上で完了です。

6.1.2 ログイン URL の設定

ここでは Tomcat がリソースを保護する際の FORM 認証をエージェントが自動で実行するために FORM 認証のログイン URL を設定します。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム →「エージェント」→「J2EE」→対象のエージェントを開きます。
2. 「アプリケーション」タブの「ログインフォーム URI」に保護するアプリケーションのログインフォームの URL を次の形式で入力します。

   [コンテキストパス]＋[web.xml の”<form-login-page>” タグで定義した URL]

   例)　コンテキストパス：　”/sampleapp”

   <form-login-page>：　”/login.html”

   ログインフォーム URI：　”/sampleapp/login.html”

3. 画面右上の「保存」をクリックして設定を保存します。

以上で完了です。

6.2 アプリケーション側の設定

保護対象の WEB アプリケーションは Tomcat で認可を行うために web.xml を編集します。

6.2.1 ロールの設定

OpenAM のグループをロールとして定義します。例では“manager”グループをロールとして定義しています。

```
<security-role>
  <role-name>id=manager,ou=group,dc=opensso,dc=java,dc=net</role-name>
</security-role>
```

6.2.2 FORM 認証の設定

WEB アプリケーションを Tomcat で認証するための設定を行います。実際にはエージェントが用意した認証ページで自動認証を行うため、リソースが存在しなくてもかまいません。

```
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/login.html</form-login-page>
    <form-error-page>/err.html</form-error-page>
  </form-login-config>
</login-config>
```

6.2.3 認可の設定

Tomcat で保護するリソースを URL 単位で定義します。ここでは全てのリソースを”manager”グループに対してのみ許可しています。

```
<web-resource-collection>
  <web-resource-name>FormAuth</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>id=manager,ou=group,dc=opensso,dc=java,dc=net</role-name>
  </auth-constraint>
</security-constraint>
```

7. 冗長構成

ポリシーエージェントがロードバランサー背後にある場合の注意事項や設定方法を説明します。

7.1 ポリシーエージェント側

ポリシーエージェント自体はロードバランサーを考慮する必要はありません。保護対象の WEB アプリケーションの仕様に従って対応してください。

8. ログ

エージェントのログには以下の 2 種類のログがあります。それぞれのログについて、出力方法を説明します。

1. デバッグログ
2. 監査ログ

8.1 デバッグログ

8.1.1 デバッグログの出力先

デバッグログはエージェントが稼働するサーバーの以下のディレクトリに出力されます。

- /opt/osstech/share/openam-agent-tomcat6/Agent_XXX/logs/debug

8.1.2 ログレベルの変更

デバッグログには以下のログレベルがあります。

- エラー
- メッセージ
- 情報
- 警告

デバッグログの出力設定は、管理コンソール(OpenAM 側)から行います。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム名」→「エージェント」→「J2EE」→「エージェント」の対象エージェントを開きます。
2. 「一般」の「エージェントデバッグレベル」からデバッグレベルを選択します。
3. 画面右上の「保存」をクリックし設定を保存します。

設定後は Tomcat を再起動してください。

8.2 監査ログ

8.2.1 監査ログの出力設定

監査ログを出力するための設定手順は以下のようになります。

1. OpenAM 管理コンソールの「アクセス制御」→「レルム」の対象レルム→「エージェント」→「J2EE」を開きます。
2. 「エージェント」の一覧から、対象のエージェントをクリックします。
3. 「監査」の「監査アクセスタイプ」から監査ログの出力方式を選択します。
   a) 「LOG_ALLOW」：アクセスが許可された場合にログ出力します。
   b) 「LOG_DENY」：アクセスが拒否された場合にログ出力します。
   c) 「LOG_BOTH」：アクセスが許可された場合と拒否された場合にログ出力します。
   d) 「LOG_NONE」：ログを出力しません。
4. 「監査ログ位置」からログの出力先を選択します。
   a) 「すべて」：OpenAM サーバーとエージェントが稼働するサーバー両方にログが出力されます。
   b) 「リモート」：OpenAM サーバーにログが出力されます。

      OpenAM サーバーの監査ログの出力先は下記のディレクトリです。

      - /opt/osstech/share/tomcat6/openam/openam/log

   c) 「ローカル」：エージェントが稼働するサーバーにログが出力されます。

      エージェントが稼働するサーバーの監査ログの出力先は下記のディレクトリです。

      - /opt/osstech/share/openam-agent-tomcat6/Agent_XXX/logs/audit
5. 画面右上の「保存」をクリックし設定を保存します。

設定後は Tomcat を再起動してください。

9. トラブルシューティング

ポリシーエージェントでエラーが発生した場合、デバッグログを確認してください。デバッグログの出力先については『8.1.1 デバッグログの出力先』を参照してください。

有効なログが出力されていない場合はログレベルの見直しが必要です。『8.1.2 ログレベルの変更』を参照してログレベルを変更し、エラーを再現してください。

また、OpenAM のポリシーにより WEB アプリケーションへのアクセスが拒否されている場合は監査ログも確認してください。監査ログの出力先については『8.2.1 監査ログの出力設定』を参照してください。

9.1 デバッグログ

9.1.1 ログフォーマット

デバッグログのサンプルを以下に示します。

```
amSecurity:09/03/2012 07:46:10:670 PM JST: Thread[main,5,main]
ERROR: AdminTokenAction: FATAL ERROR: Cannot obtain Application SSO token.
Check AMConfig.properties for the following properties
   com.sun.identity.agents.app.username
   com.iplanet.am.service.password
```

9.2 監査ログ

9.2.1 ログフォーマット

監査ログのサンプルを以下に示します。

OpenAM サーバーにログを出力する場合

1行目はログの項目を示しています。最初のログがアクセスが許可されたケース、次のログがアクセスが拒否されたケースです。

```
#Fields: time Data LoginID ContextID IPAddr LogLevel Domain LoggedBy MessageID ModuleName NameID
HostName
"2012-09-03 19:23:33" "Access to http://agent.example.co.jp:8080/agentsample/jsp/showHttpHeaders.jsp
allowed for user id=amadmin,ou=user,dc=opensso,dc=java,dc=net"
id=amadmin,ou=user,dc=opensso,dc=java,dc=net 8388ee57274f1ddc01 null INFO dc=opensso,dc=java,dc=net
id=tomcat6,ou=agent,dc=opensso,dc=java,dc=net "Not Available" amAgent_agent_example_co_jp_8080.log
"Not Available" 172.16.43.1
"2012-09-03 19:24:06" "Access to http://agent.example.co.jp:8080/agentsample/jsp/showHttpHeaders.jsp
denied for user id=demo,ou=user,dc=opensso,dc=java,dc=net" id=demo,ou=user,dc=opensso,dc=java,dc=net
96b483c837bb5ca301 null INFO dc=opensso,dc=java,dc=net id=tomcat6,ou=agent,dc=opensso,dc=java,dc=net
"Not Available" amAgent_apsrv2_example_co_jp_8080.log "Not Available" 172.16.43.1
```

OpenAM のログの項目は以下の通りです。

項目	内容
time	時間
Data	ログの概要
LoginID	ログインユーザーの ID
ContextID	セッションに関連付けられた ID
IPAddr	IP アドレス
LogLevel	ログレベル
Domain	ドメイン
LoggedBy	ログを出力したユーザーの ID
MessageID	メッセージに関連付けられた ID
ModuleName	モジュール名
NameID	NameID
HostName	ホスト名

エージェントが稼動するサーバーにログを出力する場合

最初のログがアクセスが許可されたケース、次のログがアクセスが拒否されたケースです。

```
[Monday, September 3, 2012 7:23:33 PM JST] Access to
http://agent.example.co.jp:8080/agentsample/jsp/showHttpHeaders.jsp allowed for user
id=amadmin,ou=user,dc=opensso,dc=java,dc=net
[Monday, September 3, 2012 7:24:06 PM JST] Access to
http://apsrv2.example.co.jp:8080/agentsample/jsp/showHttpHeaders.jsp denied for user
id=demo,ou=user,dc=opensso,dc=java,dc=net
```

10. 改版履歴

- 2012 年 9 月 7 日　　リビジョン 1.0
  - 初版作成