OpenAM Web Policy Agent User's Guide

Version 4

Mark Craig
Gene Hirayama
Mike Jang
Chris Lee
Vanessa Richie

ForgeRock AS
201 Mission St., Suite 2900
San Francisco, CA 94105, USA
+1 415-599-1100 (US)
www.forgerock.com

Abstract

Guide to installing OpenAM web policy agents. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software.

Table of Contents

Preface

This guide shows you how to install OpenAM web server policy agents, as well as how to integrate with other access management software. Read the OpenAM Web Policy Agent Release Notes before you get started.

1 Who Should Use This Guide

This guide is written for anyone installing OpenAM policy agents to interface with supported web servers application containers.

This guide covers procedures that you theoretically perform only once per version. This guide aims to provide you with at least some idea of what happens behind the scenes when you perform the steps.

You do not need to be an OpenAM wizard to learn something from this guide, though a background in access management and maintaining web application software can help. You do need some background in managing services on your operating systems and in your application servers. You can nevertheless get started with this guide, and then learn more as you go along.

2 Formatting Conventions

Most examples in the documentation are created in GNU/Linux or Mac OS X operating environments. If distinctions are necessary between operating environments, examples are labeled with the operating environment name in parentheses. To avoid repetition file system directory names are often given only in UNIX format as in /path/to/server, even if the text applies to C:\path\to\server as well.

Absolute path names usually begin with the placeholder /path/to/. This path might translate to /opt/, C:\Program Files\, or somewhere else on your system.

Command-line, terminal sessions are formatted as follows:

```
$ echo $JAVA_HOME
/path/to/jdk
```

Command output is sometimes formatted for narrower, more readable output even though formatting parameters are not shown in the command. In the following example, the query string parameter _prettyPrint=true is omitted and some of the output is replaced with an ellipsis (...):

```
$ curl https://bjensen:hifalutin@opendj.example.com:8443/users/newuser
{
  "_rev" : "000000005b337348",
  "_id" : "newuser",
  ...
}
```

Program listings are formatted as follows:

```
class Test {
    public static void main(String [] args)  {
        System.out.println("This is a program listing.");
    }
}
```

3 Accessing Documentation Online

ForgeRock core documentation, such as this document, aims to be technically accurate and complete with respect to the software documented.

Core documentation therefore follows a three-phase review process designed to eliminate errors:

- Product managers and software architects review project documentation design with respect to the readers' software lifecycle needs.

- Subject matter experts review proposed documentation changes for technical accuracy and completeness with respect to the corresponding software.

- Quality experts validate implemented documentation changes for technical accuracy, completeness in scope, and usability for the readership.

The review process helps to ensure that documentation published for a ForgeRock release is technically accurate and complete.

Fully reviewed, published core documentation is available at http://backstage.forgerock.com/. Use this documentation when working with a ForgeRock Enterprise release.

You can find pre-release draft documentation at the online community resource center. Use this documentation when trying a nightly build.

4 Joining the ForgeRock Community

Visit the Community resource center where you can find information about each project, download nightly builds, browse the resource catalog, ask and answer questions on the forums, find community events near you, and of course get the source code as well.

5 Getting Support and Contacting ForgeRock

ForgeRock provides support services, professional services, classes through ForgeRock University, and partner services to assist you in setting up and maintaining your deployments. For a general overview of these services, see https://www.forgerock.com.

ForgeRock has staff members around the globe who support our international customers and partners. If you have any questions, contact ForgeRock using the address or telephone number nearest to you. Find the latest addresses and telephone numbers at https://www.forgerock.com, or send an email to ForgeRock at info@forgerock.com.

Chapter 1

About OpenAM Web Policy Agents

OpenAM web policy agents provide light touch integration for web applications running on supported web servers. This chapter covers what web policy agents do and how they work.

A *policy agent* enforces policy for OpenAM and protects all resources on the web server. The policy agent intercepts requests from users trying to access a protected web resource and denies access until the user has authorization from OpenAM to access the resource.

Note

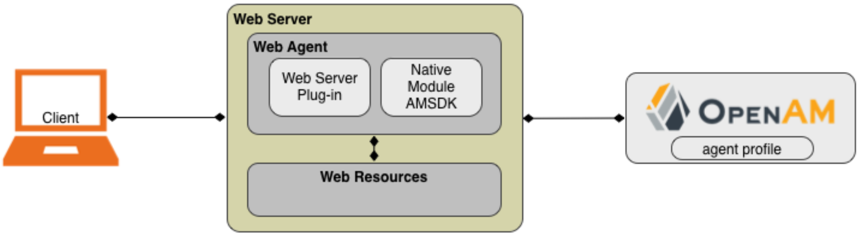
A single policy agent can work with multiple applications. You therefore install only one policy agent per web server.

Installing more than one policy agent in a web server is not supported.

1.1 Web Policy Agent Components

The web policy agent provides fast installation and light touch integration to protect the resources on the supported web server. The web agent consists of a web server plugin matching the API requirements of the particular web server and a native module that interfaces with OpenAM for its services.

Figure 1.1. Web Policy Agent



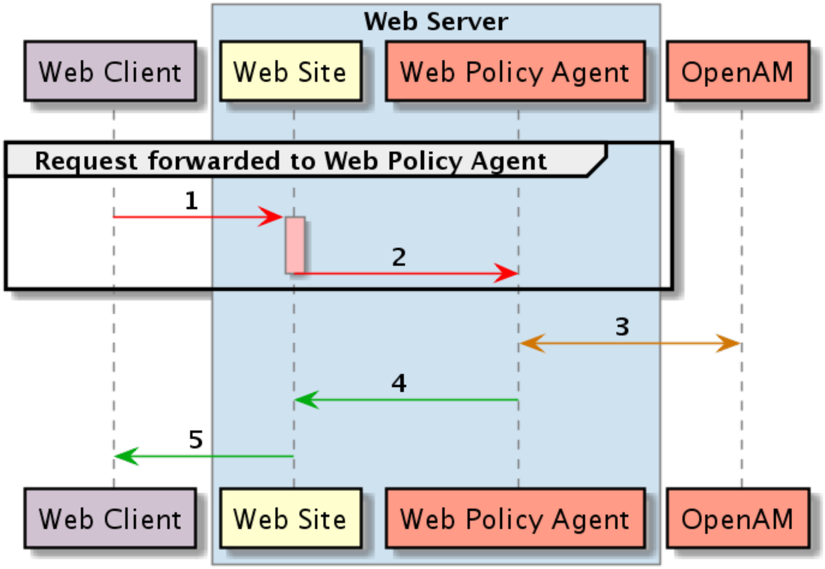
1.2 How the User, Web Policy Agent, and OpenAM Interact

Imagine that a user attempts to access a protected resource before having authenticated by pointing the user's browser to a web page. Assume that you have configured OpenAM to protect the web page. Then, the web policy agent intercepting the user's browser's request finds no session token in the request, and so redirects the user's browser to the OpenAM login page for authentication. After the user has successfully authenticated, OpenAM sets a session token in a browser cookie, and redirects the browser back to the page the user tried to access initially.

When the user's browser reiterates the request, the policy agent again checks that the request has a session token, finds a session token this time, and validates the session token with OpenAM. Given the valid session token, the policy agent gets a policy decision from OpenAM concerning whether the user can access the page. If OpenAM's Policy Service determines that the user is allowed to access the page, OpenAM responds to the policy agent that access should be granted. The web policy agent then permits the web page to be returned to the user's browser.

The following diagram shows how the pieces fit together when a web client accesses a web page protected by a policy agent. This diagram is simplified to show only the essential principals rather than to describe every possible case.

Figure 1.2. Web Policy Agent Interaction



A web policy agent is a library installed in the web server and configured to be called by the web server when a client requests access to a protected resource in a web site. Here is how it works:

1. The web client requests access to a protected resource.

2. The web server runs the request through the policy agent that protects the resource according to OpenAM policy. The policy agent acts to enforce policy, whereas the policy configuration and decisions are handled by OpenAM.

3. The policy agent communicates with OpenAM to get the policy decision to enforce.

4. For a resource to which OpenAM approves access, the policy agent allows access.

5. The web server returns the requested access to the web client.

Chapter 2

Web Policy Agent Features

The Web policy agent provides a number of additional features useful for your deployment, some of which are described below.

2.1 Multiple Sites and Virtual Host Support

Web policy agent instances can be configured to operate with multiple websites in IIS, and with multiple virtual hosts in Apache.

Each configuration instance is independent and has its own configuration file, debug logs, and audit logs. Each instance can connect to a different OpenAM realm, or even different OpenAM servers.

For more information, see Section 4.3, "Installing Apache Web Policy Agents into a Virtual Host" and Section 5.2, "Installing IIS Web Policy Agents".

2.2 Web Agent SSO Only Mode

The agent intercepts all inbound client requests to access a protected resource and processes the request based on a global configuration property, `com.sun.identity.agents.config.sso.only`. The configuration setting determines the mode of operation that should be carried out on the intercepted inbound request.

When `com.sun.identity.agents.config.sso.only` is `true`, the web policy agent only manages user authentication. The filter invokes the OpenAM Authentication service to verify the identity of the user. If the user's identity is verified, the user is issued a session token through OpenAM's Session service.

When `com.sun.identity.agents.config.sso.only` is `false`, which is the default, the web policy agents will also manage user authorization, by using the policy engine in OpenAM.

For more information, see Section 3.6.3, "Configuring Web Policy Agent SSO Properties".

2.3 Not-Enforced URL and Client IP Lists

The policy agent supports properties to bypass authentication and grant immediate access to resources not requiring protection, such as images, stylesheets, or static HTML pages.

You can configure a Not-Enforced URL List using the `com.sun.identity.agents.config.notenforced.url` property that grants the user access to resources whose URLs match those in the list.

For example, you can set URL patterns with wildcards in the OpenAM console using the following patterns:

```
/logout.html
/images/*
/css/-*-
/*.jsp?locale=*
```

For more information on wildcard usage, see Wildcard Usage.

The policy agent supports a Not-Enforced Client IP List, which specifies the client IP addresses that can be excluded from authentication and authorization. This property is useful to allow administrators access to the web site from a certain IP address or allow a search engine access to the web resources.

For finer control, you can configure a not-enforced policy that applies to requests to specified URLs, which also come from a list of specified IP addresses. See Not Enforced URL from IP Processing Properties (Not yet in OpenAM console).

For more information on not-enforced lists, see Section 3.6.2, "Configuring Web Policy Agent Application Properties".

2.4 Attribute Fetch Modes

Web policy agents provide the capability to fetch and inject user information into HTTP headers, request objects, and cookies and pass them on to the protected client applications. The client applications can then personalize content using these attributes in their web pages or responses.

Specifically, you can configure the type of attributes to be fetched and the associated mappings for the attributes names used on OpenAM to those values used in the containers. The web policy agent securely fetches the user and session data from the authenticated user as well as policy response attributes.

For example, you can have a web page that addresses the user by name retrieved from the user profile, for example "Welcome Your Name!" OpenAM populates part of the request (header, form data) with the CN from the user profile, and the web site consumes and displays it.

For more details, see Profile Attributes Processing Properties.

2.5 FQDN Checking

The web policy agent requires that clients accessing protected resources use valid URLs with fully qualified domain names (FQDNs). If invalid URLs are referenced, policy evaluation can fail as the FQDN will not match the requested URL, leading to blocked access to the resource. Misconfigured URLs can also result in incorrect policy evaluation for subsequent access requests.

There are cases where clients may specify resource URLs that differ from the FQDNs stored in OpenAM policies, for example, in load balanced and virtual host environments. To handle these cases, the web policy agent supports FQDN Checking properties: `FQDN Default` and `FQDN Virtual Host Map` properties.

The `FQDN Default` property specifies the default URL with valid hostname. The property ensures that the policy agent can redirect to a URL with a valid hostname should it discover an invalid URL in the client request.

The `FQDN Virtual Host Map` property stores map keys and their corresponding values, allowing invalid URLs, load balanced URLs, and virtual host URLs to be correctly mapped to valid URLs. Each entry in the Map has precedence over the `FQDN Default` setting, so that if no valid URLs exist in the `FQDN Virtual Host Map` property, the agent redirects to the value specified in the `FQDN Default` property.

If you want the agent to redirect to a URL other than the one specified in the `FQDN Default` property, then it is good practice to include any anticipated invalid URLs in the `FQDN Virtual Host Map` property and map it to a valid URL.

For more details, see Fully Qualified Domain Name Checking Properties.

2.6 Cookie Reset Properties

OpenAM provides cookie reset properties that the agent carries out prior to redirecting the client to a login page for authentication.

Cookie reset is typically used when multiple parallel authentication mechanisms are in play with the policy agent and another authentication system. The policy agent can reset the cookies set by the other mechanism before redirecting the client to a login page.

The cookie reset properties include a name list specifying all of the cookies that will reset, a domain map specifying the domains set for each cookie, and a path map specifying the path from which the cookie will be reset.

If you have enabled attribute fetching using cookies to retrieve user data, it is good practice to use cookie reset, which will reset once you want to access an enforced URL without a valid session.

For more details, see Cookie Reset Properties.

2.7 Cross Domain Single Sign-On

Cross domain single sign-on (CDSSO) allows the web policy agent to transfer a validated stateful session ID between an OpenAM domain and an application domain using a proprietary OpenAM mechanism. Normally, single sign-on cannot be implemented across domains as the session cookie from one domain (for example, website.com) is not accessible from another domain (for example, website.net).

OpenAM's CDSSO solves this cross-domain problem and is best implemented in environments where all the domains are managed by the same organization, and where the OpenAM server is configured to use stateful sessions. OpenAM does not support CDSSO for deployments with stateless sessions.

The web policy agent works with an OpenAM component called a `CDCServlet` that generates a self-submitting form containing the valid session token from one domain. The form gets auto-submitted to the policy agent endpoint via a POST operation. The policy agent processes the request and extracts the session ID, which is again validated by OpenAM. If validation is successful, the policy agent sets the cookie in alternate domain. The client can then access a resource in that domain.

For more details, see *Configuring Cross Domain Single Sign-On*.

2.8 Supporting Load Balancers

The web policy agent provides a number of advanced properties for load balancer deployments fronting multiple policy agents. Properties are available to get the client IP and host name from the load balancer.

If the policy agent is running behind a load balancer, you can configure the policy agent to set a sticky cookie or a query parameter in the URL to ensure subsequent requests are routed to the same instance to preserve session data.

These mechanisms ensure that unauthenticated POST data can be preserved. Policy agents store POST data in the cache and do not share the data among the agents behind the load balancer.

For more details, see Section 3.7, "Configuring Web Policy Agents Behind Load Balancers".

Note

Web policy agents support more than one agent instance running on the same host by properly initializing the multi-process locks/semaphores during the bootstrap process.

Chapter 3

Configuring Web Policy Agents

You install policy agents in web servers and web application containers to enforce access policies OpenAM applies to protected web sites and web applications. Policy agents depend on OpenAM for all authentication and authorization decisions. Their primary responsibility consists in enforcing what OpenAM decides in a way that is unobtrusive to the user. In organizations with many servers, you might well install many policy agents.

Policy agent configuration is distinct from policy configuration. The only policy-like configurations that you apply to policy agents are:

- URLs to exclude from policy enforcement (*not enforced URLs*)
- Client IP addresses to exclude from policy enforcement (*not enforced IPs*)

3.1 Configuration Location

Policy agent configuration properties are either stored:

- Centrally, in the OpenAM configuration store
- Locally, as a flat file

3.1.1 Centrally Stored Agent Configuration

By default, policy agent configuration settings are stored centrally in the OpenAM configuration store. Storing the policy agent configuration centrally allows you to configure your policy agents by using the OpenAM console, the **ssoadm** command line tool, or the REST API for easier management. Any property change made in OpenAM is immediately communicated to the agent by using a notification. Many policy agent properties are hot-swap enabled, allowing the change to take effect immediately without restarting the policy agent.

You configure policy agents in realms. To access the centralized web policy agent configuration, select Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* in the OpenAM console.

For more information on creating centrally-stored agent profiles, see Section 3.4, "Creating Agent Profiles".

3.1.2 Locally Stored Agent Configuration

The policy agent installer can create a flat file with the agent configuration. The file is named `agent.conf` and is stored at the path `/web_agents/`*`agent_version`*`instances/Agent_`*`nnn`*`/config`.

If you choose to use a locally-stored agent configuration, you make all configuration changes by modifying property values in the `agent.conf` file. You cannot make changes using the OpenAM console, command-line interface, or REST API.

When using a locally-stored agent configuration, provide valid values for configuration properties ending in the following strings:

- `.cookie.name`
- `.fqdn.default`
- `.agenturi.prefix`
- `.naming.url`
- `.login.url`
- `.instance.name`
- `.username`
- `.password`
- `.connection_timeout`
- `.policy_clock_skew`

The web policy agent installer populates properties required to connect to an OpenAM instance. Additional properties are needed when settings are stored locally.

3.2 OpenIG or Policy Agent?

OpenAM supports both OpenIG and also a variety of policy agents. OpenIG and the policy agents can both enforce policy, redirecting users to authenticate when necessary, and controlling access to protected resources. OpenIG runs as a self-contained reverse proxy located between the users and the protected applications. Policy agents are installed into the servers where applications run, intercepting requests in that context.

Use OpenIG to protect access to applications not suited for a policy agent. Not all web servers and Java EE applications have policy agents. Not all operating systems work with policy agents.

Policy agents have the advantage of sitting within your existing server infrastructure. Once you have agents installed into the servers with web applications or sites to protect, then you can manage their configurations centrally from OpenAM.

For organizations with both servers on which you can install policy agents and also applications that you must protect without touching the server, you can use policy agents on the former and OpenIG for the latter.

3.3 Types of Agent

When you open the OpenAM console to configure agents for the top-level realm, you can choose from a number of different types of agents.

Each agent type requires an *agent profile* in OpenAM. The agent profile contains essential configuration for agent operation, such as a password to authenticate the agent, and the URL the agent resides at. For agents that support it, the agent profile can store all agent configuration centrally, rather than locally on the agent server.

Web and J2EE policy agents are the most common, requiring the least integration effort. The available agent types are:

Web

  You install web agents in web servers to protect web sites.

J2EE

  You install J2EE agents in web application containers to protect web applications.

Web Service Provider
: WSP agents are for use with Web Services Security.

Web Service Client
: WSC agents are for use with Web Services Security.

STS Client
: The Security Token Service client agent is for securing requests to the Security Token Service.

2.2 Agents
: Version 2.2 web and Java EE policy agents hold their configuration locally, connecting to OpenAM with a username/password combination. This agent type is provided for backwards compatibility.

OAuth 2.0 Client Agent
: OAuth 2.0 clients are registered using this type of policy agent profile.

Agent Authenticator
: The agent authenticator can read agent profiles by connecting to OpenAM with a user name, password combination, but unlike the agent profile administrator, cannot change agent configuration.

3.4 Creating Agent Profiles

This section concerns creating agent profiles, and creating groups that let agents inherit settings when you have many agents with nearly the same profile settings.

Procedure 3.1. To Create an Agent Profile

To create a new Web or Java EE policy agent profile, you need to create a name and password for the agent. You also need the URLs to OpenAM and the application to protect:

1. Login to OpenAM Console as an administrative user.

2. On the Realms menu of the OpenAM console, select the realm in which the agent profile is to be managed.

3. Click the Agents link, click the tab page for the kind of agent profile you want to create, and then click the New button in the Agent table.

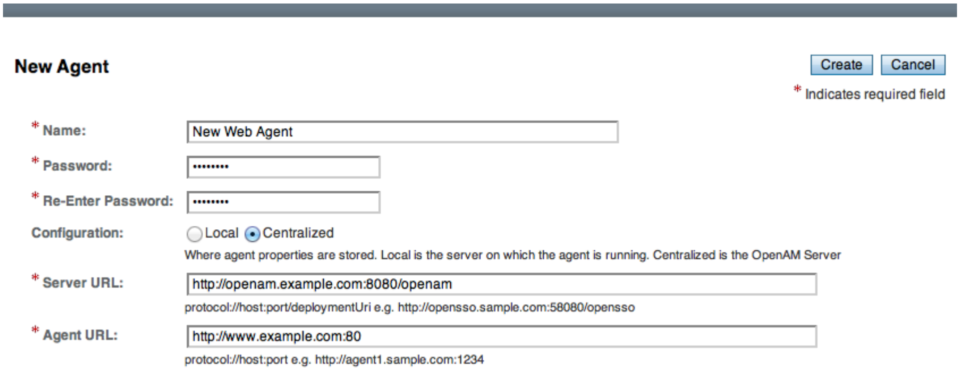
4. In the Name field, enter a name for the agent profile.

5. In the Password and Re-Enter Password fields, enter a password for the new agent profile.

6. Click `Local` or `Centralized` (Default) to determine where the agent properties are stored. If you select `Local`, the properties are stored on the server on which the agent is running. If you select `Centralized`, the properties are stored on the OpenAM server.

7. In the Server URL field, enter the URL to OpenAM. For example, `http://openam.example.com:8080/openam`.

8. In the Agent URL field, enter the primary URL of the web or application server protected by the policy agent. Note for web agents, an example URL would look like: `http://www.example.com:80`. For Java EE policy agents, an example URL must include the `agentapp` context: `http://www.example.com:8080/agentapp`.



New Agent

* Indicates required field

* Name: New Web Agent

* Password: ••••••••

* Re-Enter Password: ••••••••

Configuration: ○ Local ● Centralized

Where agent properties are stored. Local is the server on which the agent is running. Centralized is the OpenAM Server

* Server URL: http://openam.example.com:8080/openam

protocol://host:port/deploymentUri e.g. http://opensso.sample.com:58080/opensso

* Agent URL: http://www.example.com:80

protocol://host:port e.g. http://agent1.sample.com:1234

9. Click Create. After creating the agent profile, you can click the link to the new profile to adjust and export the configuration.

Procedure 3.2. To Create an Agent Profile Group and Inherit Settings

Agent profile groups let you set up multiple agents to inherit settings from the group. To create a new agent profile group, you need a name and the URL to the OpenAM server in which you store the profile:

1. Login to OpenAM Console as an administrative user.

2. On the Realms menu of the OpenAM console, Select the realm in which you manage agents.

3. Click the Agents link, click the tab page for the kind of agent group you want to create, and then in the Group table, click New.

   After creating the group profile, you can click the link to the new group profile to fine-tune or export the configuration.

4. Inherit group settings by selecting your agent profile, and then selecting the group name in the Group drop-down list near the top of the profile page.

   You can then adjust inheritance by clicking Inheritance Settings on the agent profile page.

Procedure 3.3. To Create an Agent Profile Using the Command Line

You can create a policy agent profile in OpenAM using the **ssoadm** command-line tool. You do so by specifying the agent properties either as a list of attributes, or by using an agent properties file as shown below. Export an existing policy agent configuration before you start to see what properties you want to set when creating the agent profile.

The following procedure demonstrates creating a policy agent profile using the **ssoadm** command:

1. Make sure the **ssoadm** command is installed. See "To Set Up Administration Tools" in the *OpenAM Installation Guide*.

2. Determine the list of properties to set in the agent profile.

   The following properties file shows a minimal configuration for a policy agent profile:

   ```
   $ cat myAgent.properties
   com.sun.identity.agents.config.agenturi.prefix=http://www.example.com:80/amagent
   com.sun.identity.agents.config.cdsso.cdcservlet.url[0]= \
        https://openam.example.com:8443/openam/cdcservlet
   com.sun.identity.agents.config.fqdn.default=www.example.com
   com.sun.identity.agents.config.login.url[0]= \
        http://openam.example.com:8443/openam/UI/Login
   com.sun.identity.agents.config.logout.url[0]= \
        http://openam.example.com:8443/openam/UI/Logout
   com.sun.identity.agents.config.remote.logfile=amAgent_www_example_com_80.log
   com.sun.identity.agents.config.repository.location=centralized
   com.sun.identity.client.notification.url= \
        http://www.example.com:80/UpdateAgentCacheServlet?shortcircuit=false
   sunIdentityServerDeviceKeyValue[0]=agentRootURL=http://www.example.com:80/
   sunIdentityServerDeviceStatus=Active
   userpassword=password
   ```

3. Set up a password file used when authenticating to OpenAM. The password file must be read-only for the user who creates the policy agent profile, and must not be accessible to other users:

   ```
   $ echo password > /tmp/pwd.txt
   $ chmod 400 /tmp/pwd.txt
   ```

4. Create the profile in OpenAM:

```
$ ssoadm create-agent \
  --realm / \
  --agentname myAgent \
  --agenttype J2EE \
  --adminid amadmin
  --password-file /tmp/pwd.txt \
  --datafile myAgent.properties

Agent configuration was created.
```

At this point you can view the profile in OpenAM Console under Realms > *Realm Name* > Agents to make sure the configuration is what you expect.

3.5 Delegating Agent Profile Creation

If you want to create policy agent profiles when installing policy agents, then you need the credentials of an OpenAM user who can read and write agent profiles.

You can use the OpenAM administrator account when creating policy agent profiles. If you delegate policy agent installation, then you might not want to share OpenAM administrator credentials with everyone who installs policy agents.

Procedure 3.4. To Create Agent Administrators for a Realm

Follow these steps to create *agent administrator* users for a realm:

1. In OpenAM console, browse to Realms > *Realm Name* > Subjects.

2. Under Group click New... and create a group for agent administrators.

3. Switch to the Privileges tab for the realm, and click the name of the group you created.

4. Select Read and write access to all configured agents, and then Save your work.

5. Return to the Subjects tab, and under User create as many agent administrator users as needed.

6. For each agent administrator user, edit the user profile.

   Under the Group tab of the user profile, add the user to agent profile administrator group, and then Save your work.

7. Provide each system administrator who installs policy agents with their agent administrator credentials.

When installing the policy agent with the `--custom-install` option, the system administrator can choose the option to create the profile during installation, and then provide the agent administrator user name and the path to a read-only file containing the agent administrator password. For silent installs, you can add the `--acceptLicense` option to auto-accept the software license agreement.

3.6 Configuring Web Policy Agent Properties

When you create a web policy agent profile and install the agent, you can choose to store the agent configuration centrally and configure the agent through OpenAM console. Alternatively, you can choose to store the agent configuration locally and configure the agent by changing values in the properties file. This section covers centralized configuration, indicating the corresponding properties for use in a local configuration file where applicable. [1]

Some properties do not yet appear in the OpenAM Console, so they need to be configured as custom properties, see Section 3.6.7, "Configuring Web Policy Agent Custom Properties", or locally in the agent properties configuration file, `agent.conf`.

Tip

To show the agent properties in configuration file format that correspond to what you see in the console, click Export Configuration after editing agent properties.

This corresponds to the local Java properties configuration file that is set up when you install an agent, for example in `agent_1/config/agent.conf`.

After changing properties specified as "Hot swap: no", you must restart the agent's container for the changes to take effect.

3.6.1 Configuring Web Policy Agent Global Properties

This section covers global web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* > Global.

[1] The configuration file syntax is that of a standard Java properties file. See java.util.Properties.load for a description of the format. The value of a property specified multiple times is not defined.

This section describes the following property groups:

- Profile Properties
- General Properties
- Audit Properties
- Fully Qualified Domain Name Checking Properties

Profile Properties

Group
`agentgroup`
: For assigning the agent to a previously configured web agent group in order to inherit selected properties from the group.

  Property: `agentgroup`

Password
: Agent password used when creating the password file and when installing the agent.

  Property: `userpassword`

Status
: Status of the agent configuration.

  Property: `sunIdentityServerDeviceStatus`

Location of Agent Configuration Repository
: Whether the agent's configuration is managed centrally through OpenAM (`centralized`) or locally in the policy agent configuration file (`local`).

  If you change this to a local configuration, you can no longer manage the policy agent configuration through OpenAM console.

  Property: `com.sun.identity.agents.config.repository.location`

Agent Configuration Change Notification
: Enable agent to receive notification messages from OpenAM server for configuration changes.

  Property: `com.sun.identity.agents.config.change.notification.enable`

Enable Notifications
: If enabled, the agent receives policy updates from the OpenAM notification mechanism to maintain its internal cache. If disabled, the agent must poll OpenAM for changes.

  Property: `com.sun.identity.agents.config.notification.enable`

Hot swap: no

Agent Notification URL
URL used by agent to register notification listeners.

Property: `com.sun.identity.client.notification.url`

Hot swap: no

Agent Deployment URI Prefix
The default value is *`agent-root-URL`*`/amagent`.

Property: `com.sun.identity.agents.config.agenturi.prefix`

Hot swap: yes

Configuration Reload Interval
Interval in minutes to fetch agent configuration from OpenAM. Used if notifications are disabled. Default: 60.

Property: `com.sun.identity.agents.config.polling.interval`

Hot swap: no

Configuration Cleanup Interval
Interval in minutes to cleanup old agent configuration entries unless they are referenced by current requests. Default: 30.

Property: `com.sun.identity.agents.config.cleanup.interval`

Hot swap: no

Agent Root URL for CDSSO
The agent root URL for CDSSO. The valid value is in the format *`protocol`*`://`*`hostname`*`:`*`port`*`/` where *`protocol`* represents the protocol used, such as `http` or `https`, *`hostname`* represents the host name of the system where the agent resides, and *`port`* represents the port number on which the agent is installed. The slash following the port number is required.

If your agent system also has virtual host names, add URLs with the virtual host names to this list as well. OpenAM checks that the `goto` URLs match one of the agent root URLs for CDSSO.

Property: `sunIdentityServerDeviceKeyValue[0]=agentRootURL`

General Properties

SSO Only Mode
When enabled, the agent enforces authentication, so that upon verification of the user's identity, the user receives a session token.

When `true` , the web policy agent only manages user authentication. The filter invokes the OpenAM Authentication service to verify the identity of the user. If the user's identity is verified, the user is issued a session token through OpenAM's Session service.

When `false`, which is the default, the web policy agents will also manage user authorization, by using the policy engine in OpenAM.

Property: `com.sun.identity.agents.config.sso.only`

Resources Access Denied URL

The URL of the customized access denied page. If no value is specified (default), then the agent returns an HTTP status of 403 (Forbidden).

Property: `com.sun.identity.agents.config.access.denied.url`

Agent Debug Level

Default is `Error`. Increase to `Message` or even `All` for fine-grained detail.

Property: `com.sun.identity.agents.config.debug.level`

You can set the level in the `agent.conf` configuration file by module by using the format *`module`*`[:`*`level`*`][,`*`module`*`[:`*`level`*`]]*`, where *`module`* is one of `AuthService`, `NamingService`, `PolicyService`, `SessionService`, `PolicyEngine`, `ServiceEngine`, `Notification`, `PolicyAgent`, `RemoteLog`, or `all`, and *`level`* is one of the following.

- `0`: Disable logging from specified module

  At this level the agent nevertheless logs messages having the level value `always`.

- `1`: Log error messages
- `2`: Log warning and error messages
- `3`: Log info, warning, and error messages
- `4`: Log debug, info, warning, and error messages
- `5`: Like level 4, but with even more debugging messages

When you omit *`level`*, the agent uses the default level, which is the level associated with the `all` module.

The following example used in the local configuration sets the log overall level to debug for all messages.

```
com.sun.identity.agents.config.debug.level=all:5
```

Agent Debug File Rotation

When enabled, rotate the debug file when specified file size is reached.

Property: `com.sun.identity.agents.config.debug.file.rotate`

Agent Debug File Size

Debug file size in bytes beyond which the log file is rotated. The minimum is 5242880 bytes (5 MB), and lower values are reset to 5 MB. OpenAM console sets a default of 10000000 bytes (approximately 10 MB).

Tip

If `com.sun.identity.agents.config.debug.file.rotate` is enabled, setting `com.sun.identity.agents.config.debug.file.size` to `-1` in the `agent.conf` file will rotate debug log files once every 24 hours rather than at a specified size limit.

Property: `com.sun.identity.agents.config.debug.file.size`

Default: 10000000

`com.sun.identity.agents.config.local.logfile` (Not yet in OpenAM console)

Name of file stored locally on the agent that contains agent debug messages.

Default:

/web_agents/**`agent_version`**/instances/agent_**`nnn`**/logs/debug/debug.log

Audit Properties

Audit Access Types

Types of messages to log based on user URL access attempts.

Property: `com.sun.identity.agents.config.audit.accesstype`

Valid values for the configuration file property include `LOG_NONE`, `LOG_ALLOW`, `LOG_DENY`, and `LOG_BOTH`.

Audit Log Location

Specifies where audit messages are logged. By default, audit messages are logged remotely.

Property: `com.sun.identity.agents.config.log.disposition`

Valid values for the configuration file property include `REMOTE`, `LOCAL`, and `ALL`.

Remote Log Filename

Name of file stored on OpenAM server that contains agent audit messages if log location is remote or all.

Property: `com.sun.identity.agents.config.remote.logfile`

Hot swap: no

Remote Audit Log Interval

Periodic interval in minutes in which audit log messages are sent to the remote log file.

Property: `com.sun.identity.agents.config.remote.log.interval`

Default: 5

Hot swap: no

Rotate Local Audit Log

When enabled, audit log files are rotated when reaching the specified size.

Property: `com.sun.identity.agents.config.local.log.rotate`

Local Audit Log Rotation Size

Beyond this size limit in bytes, the agent rotates the local audit log file if rotation is enabled. The minimum is 5242880 bytes (5 MB), and lower values are reset to 5 MB. OpenAM console sets a default of 52428800 bytes (50 MB).

Property: `com.sun.identity.agents.config.local.log.size`

Default: 52428800

`com.sun.identity.agents.config.local.audit.logfile` (Not yet in OpenAM console)

Name of file stored locally on the agent that contains agent audit messages if log location is LOCAL or ALL.

Default:

/web_agents/**`agent_version`**/instances/agent_**`nnn`**/logs/audit/audit.log

Fully Qualified Domain Name Checking Properties

FQDN Check

Enables checking of FQDN default value and FQDN map values.

Property: `com.sun.identity.agents.config.fqdn.check.enable`

FQDN Default

FQDN that the users should use in order to access resources. Without this value, the web server can fail to start, thus you set the property on agent installation, and only change it when absolutely necessary.

This property ensures that when users access protected resources on the web server without specifying the FQDN, the agent can redirect the users to URLs containing the correct FQDN.

Property: `com.sun.identity.agents.config.fqdn.default`

FQDN Virtual Host Map

Enables virtual hosts, partial hostname, and IP address to access protected resources. Maps invalid or virtual name keys to valid FQDN values so the agent can properly redirect users and the agents receive cookies belonging to the domain.

To map `myserver` to `myserver.mydomain.example`, enter `myserver` in the Map Key field, and enter `myserver.mydomain.example` in the Corresponding Map Value field. This corresponds to `com.sun.identity.agents.config.fqdn.mapping[myserver]= myserver.mydomain.example`.

Invalid FQDN values can cause the web server to become unusable or render resources inaccessible.

Property: `com.sun.identity.agents.config.fqdn.mapping`

3.6.2 Configuring Web Policy Agent Application Properties

This section covers application web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *Realm Name* > Agents > Web > *Agent Name* > Application.

This section describes the following property groups:

- Not Enforced URL Processing Properties
- Not Enforced IP Processing Properties
- Not Enforced URL from IP Processing Properties (Not yet in OpenAM console)
- Profile Attributes Processing Properties
- Response Attributes Processing Properties
- Session Attributes Processing Properties
- Common Attributes Fetching Processing Properties

Not Enforced URL Processing Properties

Ignore Path Info for Not Enforced URLs

When enabled, the path info and query are stripped from the request URL before being compared with the URLs of the not enforced list for those URLs containing a wildcard character. This prevents a user from accessing `http://`

host/index.html by requesting http://host/index.html/hack.gif when the not enforced list includes http://host/*.gif.

For a more generally applicable setting, see Ignore Path Info Properties.

Property: com.sun.identity.agents.config.ignore.path.info.for.not.enforced.list

Enable Regular Expressions for Not Enforced URLs (Not yet in OpenAM console)

Enable use of Perl-compatible regular expressions in Not Enforced URL settings by using the following property under Advanced > Custom Properties in the agent profile.

Property: org.forgerock.agents.notenforced.url.regex.enable

Not Enforced URLs

List of URLs for which no authentication is required. You can use wildcards to define a pattern for a URL.

The * wildcard matches all characters except question mark (?), cannot be escaped, and spans multiple levels in a URL. Multiple forward slashes do not match a single forward slash, so * matches mult/iple/dirs, yet mult/*/dirs does not match mult/dirs.

The -*- wildcard matches all characters except forward slash (/) or question mark (?), and cannot be escaped. As it does not match /, -*- does not span multiple levels in a URL.

OpenAM does not let you mix * and -*- in the same URL.

Examples include http://www.example.com/logout.html, http://www.example.com/images/*, http://www.example.com/css/-*-, and http://www.example.com/*.jsp?locale=*.

Trailing forward slashes are not recognized as part of a resource name. Therefore http://www.example.com/images// and http://www.example.com/images are equivalent.

Property: com.sun.identity.agents.config.notenforced.url

If you enabled use of Perl-compatible regular expressions to match Not Enforced URLs, then all your settings must be done using regular expressions. (Do not mix settings; use either the mechanism described above or Perl-compatible regular expressions, but not both.)

The following example shows settings where no authentication is required for URLs whose path ends /PublicServletA or /PublicServletB (with or without query string parameters), and no authentication is required to access .png, .jpg, .gif, .js, or .css files under URLs that do not contain /protectedA/ or /protectedB/.

```
.*/(PublicServletA|PublicServletB)(\?.*|$)
      ^(?!.*(/protectedA/|/protectedB/)).*\.(png|jpg|gif|js|css)(\?.*|$)
```

Invert Not Enforced URLs

Only invert the Not Enforced list of URLs. In other words, enforce policy only for those URLs and patterns specified in the list.

Property: `com.sun.identity.agents.config.notenforced.url.invert`

Fetch Attributes for Not Enforced URLs

When enabled, the agent fetches profile, response, and session attributes that are mapped by doing policy evaluation, and forwards these attributes to not enforced URLs.

Property: `com.sun.identity.agents.config.notenforced.url.attributes.enable`

Not Enforced IP Processing Properties

Not Enforced Client IP List

No authentication and authorization are required for the requests coming from these client IP addresses.

Property: `com.sun.identity.agents.config.notenforced.ip`

Note

Loopback addresses are not considered valid IPs on the Not Enforced IP list. If specified, the policy agent ignores the loopback address.

CIDR Client IP Specification (Not yet in OpenAM console)

As of version 3.0.4, web policy agents with this property set to `cidr` can use IPv4 netmasks and IP ranges instead of wildcards as values for Not Enforced Client IP addresses. Version 3.0.5 adds support for IPv6, including the IPv6 loopback address, `::1`.

When the parameter is defined, wildcards are ignored in Not Enforced Client IP settings. Instead, you can use settings, such as those shown in the following examples.

Netmask Example

To disable policy agent enforcement for addresses in 192.168.1.1 to 192.168.1.255, use the following setting.

```
com.sun.identity.agents.config.notenforced.ip = 192.168.1.1/24
```

The following example shows an IPv6 address with a corresponding network mask.

```
com.sun.identity.agents.config.notenforced.ip = 2001:5c0:9168:0:0:0:0:2/128
```

Currently the policy agent stops evaluating properties after reaching an invalid netmask in the list.

IP Range Example

To disable policy agent enforcement for addresses between 192.168.1.1 to 192.168.4.3 inclusive, use the following setting.

```
com.sun.identity.agents.config.notenforced.ip = 192.168.1.1-192.168.4.3
```

The following example shows a range of IPv6 addresses. The example is displayed over two lines for formatting purposes.

```
com.sun.identity.agents.config.notenforced.ip = \
    2001:5c0:9168:0:0:0:0:1-2001:5c0:9168:0:0:0:0:2
```

Property: `com.forgerock.agents.config.notenforced.ip.handler`

Hot swap: no

Client IP Validation

When enabled, validate that the subsequent browser requests come from the same IP address that the SSO token is initially issued against.

Property: `com.sun.identity.agents.config.client.ip.validation.enable`

Not Enforced URL from IP Processing Properties (Not yet in OpenAM console)

`org.forgerock.agents.config.notenforced.ipurl`

No authentication and authorization are required for requests coming from specified client IP addresses that are requesting specified resource URLs.

Specify a list of IP addresses separated by spaces, the pipe (|) character, and a list of URLs separated by spaces.

The IP list can be specified by using either netmask or IP range notation:

Netmask Example

To specify requests coming from addresses in the range 192.168.1.1 to 192.168.1.255, use the following setting.

```
192.168.1.1/24
```

The following example shows an IPv6 address with a corresponding network mask.

```
2001:5c0:9168:0:0:0:0:2/128
```

IP Range Example

To specify requests coming from addresses in the range 192.168.1.1 to 192.168.4.3 inclusive, use the following setting.

```
192.168.1.1-192.168.4.3
```

The following example shows a range of IPv6 addresses.

```
2001:5c0:9168:0:0:0:0:1-2001:5c0:9168:0:0:0:0:2
```

The URL list can be specified by using wildcards (*) or regular expressions. To use regular expression matches in the URL list, set `org.forgerock.agents.config.notenforced.ext.regex.enable=true`. Do not mix using wildcards and regular expressions. Multiple values should be separated by space characters.

The following example will not require authentication or authorization for any requests coming from the specified IP addresses, when also requesting access to a `/reports` URL, or certain files under the `/images` URL. The example is displayed over three lines for formatting purposes.

```
org.forgerock.agents.config.notenforced.ipurl[0]= \
      10.1.2.1-10.1.2.7|/reports ^(?=.*(/images/)).*\.(png|jpg|gif)(\?.*|$)
org.forgerock.agents.config.notenforced.ext.regex.enable=true
```

`org.forgerock.agents.config.notenforced.ext.regex.enable`

Enable use of Perl-compatible regular expressions in Not Enforced URL from IP settings.

Profile Attributes Processing Properties

Profile Attribute Fetch Mode

When set to `HTTP_COOKIE` or `HTTP_HEADER`, profile attributes are introduced into the cookie or the headers, respectively.

Property: `com.sun.identity.agents.config.profile.attribute.fetch.mode`

Profile Attribute Map

Maps the profile attributes to HTTP headers for the currently authenticated user. Map keys are LDAP attribute names, and map values are HTTP header names.

To populate the value of profile attribute CN under `CUSTOM-Common-Name`, enter CN in the Map Key field, and enter `CUSTOM-Common-Name` in the Corresponding

Map Value field. This corresponds to `com.sun.identity.agents.config.profile.attribute.mapping[cn]=CUSTOM-Common-Name`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (`-`) become underscores (`_`). For example, `common-name` becomes `HTTP_COMMON_NAME`.

Property: `com.sun.identity.agents.config.profile.attribute.mapping`

Response Attributes Processing Properties

Response Attribute Fetch Mode

When set to `HTTP_COOKIE` or `HTTP_HEADER`, response attributes are introduced into the cookie or the headers, respectively.

Property: `com.sun.identity.agents.config.response.attribute.fetch.mode`

Response Attribute Map

Maps the policy response attributes to HTTP headers for the currently authenticated user. The response attribute is the attribute in the policy response to be fetched.

To populate the value of response attribute `uid` under `CUSTOM-User-Name`: enter `uid` in the Map Key field, and enter `CUSTOM-User-Name` in the Corresponding Map Value field. This corresponds to `com.sun.identity.agents.config.response.attribute.mapping[uid]=Custom-User-Name`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (`-`) become underscores (`_`). For example, `response-attr-one` becomes `HTTP_RESPONSE_ATTR_ONE`.

Property: `com.sun.identity.agents.config.response.attribute.mapping`

Session Attributes Processing Properties

Session Attribute Fetch Mode

When set to `HTTP_COOKIE` or `HTTP_HEADER`, session attributes are introduced into the cookie or the headers, respectively.

Property: `com.sun.identity.agents.config.session.attribute.fetch.mode`

Session Attribute Map

Maps session attributes to HTTP headers for the currently authenticated user. The session attribute is the attribute in the session to be fetched.

To populate the value of session attribute `UserToken` under `CUSTOM-userid`: enter `UserToken` in the Map Key field, and enter `CUSTOM-userid` in the

Corresponding Map Value field. This corresponds to `com.sun.identity.agents.config.session.attribute.mapping[UserToken] =CUSTOM-userid`.

In most cases, in a destination application where an HTTP header name shows up as a request header, it is prefixed by `HTTP_`, lower case letters become upper case, and hyphens (`-`) become underscores (`_`). For example, `success-url` becomes `HTTP_SUCCESS_URL`.

Property: `com.sun.identity.agents.config.session.attribute.mapping`

Common Attributes Fetching Processing Properties

Attribute Multi-Value Separator

Specifies separator for multiple values. Applies to all types of attributes, such as profile, session, and response attributes. Default: `|`.

Property: `com.sun.identity.agents.config.attribute.multi.value.separator`

3.6.3 Configuring Web Policy Agent SSO Properties

This section covers SSO web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* > SSO.

This section describes the following property groups:

- Cookie Properties
- Cross Domain SSO Properties
- Cookie Reset Properties

Cookie Properties

Cookie Name

Name of the SSO Token cookie used between the OpenAM server and the agent. Default: `iPlanetDirectoryPro`.

Property: `com.sun.identity.agents.config.cookie.name`

Hot swap: no

Cookie Security

When enabled, the agent marks cookies secure, sending them only if the communication channel is secure.

Property: `com.sun.identity.agents.config.cookie.secure`

Hot swap: no

HTTPOnly Cookies (Not yet in OpenAM console)

As of version 3.0.5, web policy agents with this property set to `true` mark cookies as HTTPOnly, to prevent scripts and third-party programs from accessing the cookies.

Property: `com.sun.identity.cookie.httponly`

Cross Domain SSO Properties

Cross Domain SSO

Enables Cross Domain Single Sign On (CDSSO) for OpenAM deployments that use stateful sessions. CDSSO is not supported for OpenAM deployments that use stateless sessions.

Property: `com.sun.identity.agents.config.cdsso.enable`

CDSSO Servlet URL

List of URLs of the available CDSSO controllers that the agent can use for CDSSO processing. For example, `http://openam.example.com:8080/openam/cdcservlet`.

Property: `com.sun.identity.agents.config.cdsso.cdcservlet.url`

Cookies Domain List

List of domains, such as `.example.com`, in which cookies have to be set in CDSSO. If this property is left blank, then the fully qualified domain name of the cookie for the agent server is used to set the cookie domain, meaning that a host cookie rather than a domain cookie is set.

To set the list to `.example.com`, and `.example.net` using the configuration file property, include the following.

```
com.sun.identity.agents.config.cdsso.cookie.domain[0]=.example.com
      com.sun.identity.agents.config.cdsso.cookie.domain[1]=.example.net
```

Property: `com.sun.identity.agents.config.cdsso.cookie.domain`

Cookie Reset Properties

Cookie Reset

When enabled, agent resets cookies in the response before redirecting to authentication.

Property: `com.sun.identity.agents.config.cookie.reset.enable`

Cookie Reset Name List

List of cookies in the format `*name*[=*value*][;Domain=*value*]`.

Concrete examples include the following with two list items configured.

- `LtpaToken`, corresponding to `com.sun.identity.agents.config.cookie.reset[0]=LtpaToken`. The default domain is taken from FQDN Default.
- `token=value;Domain=subdomain.domain.com`, corresponding to `com.sun.identity.agents.config.cookie.reset[1]= token=value;Domain=subdomain.domain.com`

Property: `com.sun.identity.agents.config.cookie.reset`

3.6.4 Configuring Web Policy Agent OpenAM Services Properties

This section covers OpenAM services web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* > OpenAM Services.

This section describes the following property groups:

- Login URL Properties
- Logout URL Properties
- Agent Logout URL Properties
- Policy Client Service Properties

Login URL Properties

OpenAM Login URL

OpenAM login page URL, such as `http://openam.example.com:8080/openam/UI/Login`, to which the agent redirects incoming users without sufficient credentials so that they can authenticate.

Property: `com.sun.identity.agents.config.login.url`

OpenAM Conditional Login URL (Not yet in OpenAM console)

To conditionally redirect users based on the incoming request URL, set this property.

This takes the incoming request domain to match, a vertical bar (|), and then a comma-separated list of URLs to which to redirect incoming users.

If the domain before the vertical bar matches an incoming request URL, then the policy agent uses the list of URLs to determine how to redirect the user-agent. If the global property FQDN Check (`com.sun.identity.agents.config.fqdn.check.enable`) is enabled for the policy agent, then the policy agent iterates through the list until it finds an appropriate redirect URL that matches the FQDN check. Otherwise, the policy agent redirects the user-agent to the first URL in the list.

Property: `com.forgerock.agents.conditional.login.url`

Examples: `com.forgerock.agents.conditional.login.url[0]= login.example.com|http://openam1.example.com/openam/UI/Login, http://openam2.example.com/openam/UI/Login`, `com.forgerock.agents.conditional.login.url[1]= signin.example.com|http://openam3.example.com/openam/UI/Login, http://openam4.example.com/openam/UI/Login`

If CDSSO is enabled for the policy agent, then this property takes CDSSO Servlet URLs for its values (`com.sun.identity.agents.config.cdsso.cdcservlet.url`), rather than OpenAM login URLs.

CDSSO examples: `com.forgerock.agents.conditional.login.url[0]= login.example.com|http://openam1.example.com/openam/cdcservlet, http://openam2.example.com/openam/cdcservlet`, `com.forgerock.agents.conditional.login.url[1]= signin.example.com|http://openam3.example.com/openam/cdcservlet, http://openam4.example.com/openam/cdcservlet`

Agent Connection Timeout
: Timeout period in seconds for an agent connection with OpenAM auth server.

Property: `com.sun.identity.agents.config.auth.connection.timeout`

Default: 2

Polling Period for Primary Server
: Interval in minutes, agent polls to check the primary server is up and running. Default: 5.

Property: `com.sun.identity.agents.config.poll.primary.server`

Hot swap: no

Logout URL Properties

OpenAM Logout URL
: OpenAM logout page URL, such as `http://openam.example.com:8080/openam/UI/Logout`.

Property: `com.sun.identity.agents.config.logout.url`

Enable Logout URL Redirect (Not yet in OpenAM console)
: Logout URL redirect is enabled by default.

When this is disabled, instead of redirecting the user-agent, the policy agent performs session logout in the background and then continues processing access to the current URL. Disable this using Advanced > Custom Properties in the agent profile.

Property: `com.forgerock.agents.config.logout.redirect.disable`

Agent Logout URL Properties

Logout URL List

List of application logout URLs, such as `http://www.example.com/logout.html`. The user is logged out of the OpenAM session when these URLs are accessed. When using this property, specify a value for the Logout Redirect URL property.

Property: `com.sun.identity.agents.config.agent.logout.url`

Agent Logout URL Regular Expression (Not yet in OpenAM console)

Perl-compatible regular expression that matches logout URLs. Set this using Advanced > Custom Properties in the agent profile.

For example, to match URLs with `protectedA` or `protectedB` in the path and `op=logout` in the query string, use the following setting:

```
com.forgerock.agents.agent.logout.url.regex= \
      .*(/protectedA\?|/protectedB\?/).*(\&op=logout\&)(.*|$)
```

When you use this property, the agent ignores the settings for Logout URL List.

Logout Cookies List for Reset

Cookies to be reset upon logout in the same format as the cookie reset list.

Property: `com.sun.identity.agents.config.logout.cookie.reset`

Logout Redirect URL

User gets redirected to this URL after logout. Specify this property alongside a Logout URL List.

Property: `com.sun.identity.agents.config.logout.redirect.url`

Policy Client Service Properties

Policy Cache Polling Period

Polling interval in minutes during which an entry remains valid after being added to the agent's cache.

Property: `com.sun.identity.agents.config.policy.cache.polling.interval`

Hot swap: no

SSO Cache Polling Period

Polling interval in minutes during which an SSO entry remains valid after being added to the agent's cache.

Property: `com.sun.identity.agents.config.sso.cache.polling.interval`

Hot swap: no

User ID Parameter
: Agent sets this value for User Id passed in the session from OpenAM to the `REMOTE_USER` server variable. Default: `UserToken`.

Property: `com.sun.identity.agents.config.userid.param`

User ID Parameter Type
: User ID can be fetched from either SESSION or LDAP attributes. Default: `SESSION`.

Property: `com.sun.identity.agents.config.userid.param.type`

Fetch Policies From The Root Resource
: When enabled, the agent caches the policy decision of the resource and all resources from the root of the resource down. For example, if the resource is `http://host/a/b/c`, then the root of the resource is `http://host/`. This setting can be useful when a client is expect to access multiple resources on the same path. Yet, caching can be expensive if very many policies are defined for the root resource.

Property: `com.sun.identity.agents.config.fetch.from.root.resource`

Default: false

Hot swap: no

Retrieve Client Hostname
: When enabled, get the client hostname through DNS reverse lookup for use in policy evaluation. This setting can impact performance.

Property: `com.sun.identity.agents.config.get.client.host.name`

Policy Clock Skew
: Time in seconds used adjust time difference between agent system and OpenAM. Clock skew in seconds = AgentTime - OpenAMServerTime.

Use this property to adjust for small time differences encountered despite use of a time-synchronization service. When this property is not set and agent time is greater than OpenAM server time, the agent can make policy calls to the OpenAM server before the policy subject cache has expired, or you can see infinite redirection occur.

Property: `com.sun.identity.agents.config.policy.clock.skew`

Hot swap: no

Realm
: Realm where OpenAM starts policy evaluation for this policy agent.

Default: / (top-level realm)

Edit this property when OpenAM should start policy evaluation in a realm other than the top-level realm, `/`, when handling policy decision requests from this policy agent.

This property is recognized by OpenAM, not the policy agent.

Property: `org.forgerock.openam.agents.config.policy.evaluation.realm`

Hot swap: yes

Application

Application where OpenAM looks for policies to evaluate for this policy agent.

Default: `iPlanetAMWebAgentService`

Edit this property when OpenAM should look for policies that belong to an application other than `iPlanetAMWebAgentService` when handling policy decision requests from this policy agent.

This property is recognized by OpenAM, not the policy agent.

Property: `org.forgerock.openam.agents.config.policy.evaluation.application`

Hot swap: yes

3.6.5 Configuring Web Policy Agent Miscellaneous Properties

This section covers miscellaneous web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *Realm Name* > Agents > Web > *Agent Name* > Miscellaneous.

This section describes the following property groups:

- Advice Handling Properties
- Locale Properties
- Anonymous user Properties
- Cookie Processing Properties
- URL Handling Properties
- Ignore Naming URL Properties
- Invalid URL properties (Not yet in OpenAM console)
- Ignore Server Check Properties
- Ignore Path Info Properties

- Multi-Byte Enable Properties
- Goto Parameter Name Properties
- Deprecated Agent Properties

Advice Handling Properties

Composite Advice Handling (Not yet in OpenAM console)

As of version 3.0.4, when set to `true`, the agent sends composite advice in the query (GET request) instead of sending it through a POST request.

Property: `com.sun.am.use_redirect_for_advice`

Locale Properties

Agent Locale

The default locale for the agent.

Property: `com.sun.identity.agents.config.locale`

Hot swap: no

Anonymous user Properties

Anonymous User

Enable or disable REMOTE_USER processing for anonymous users.

Property: `com.sun.identity.agents.config.anonymous.user.enable`

Cookie Processing Properties

Encode special characters in Cookies

When enabled, use URL encoding for special characters in cookies. This is useful when profile, session, and response attributes contain special characters, and the attributes fetch mode is set to `HTTP_COOKIE`.

Property: `com.sun.identity.agents.config.encode.cookie.special.chars.enable`

Profile Attributes Cookie Prefix

Sets cookie prefix in the attributes headers. Default: `HTTP_`.

Property: `com.sun.identity.agents.config.profile.attribute.cookie.prefix`

Profile Attributes Cookie Maxage

Maximum age in seconds of custom cookie headers. Default: 300.

Property: `com.sun.identity.agents.config.profile.attribute.cookie.maxage`

URL Handling Properties

URL Comparison Case Sensitivity Check

When enabled, enforces case insensitivity in both policy and not enforced URL evaluation.

Property: `com.sun.identity.agents.config.url.comparison.case.ignore`

Encode URL's Special Characters

When enabled, encodes the URL which has special characters before doing policy evaluation.

Property: `com.sun.identity.agents.config.encode.url.special.chars.enable`

Ignore Naming URL Properties

Ignore Preferred Naming URL in Naming Request

When enabled, do not send a preferred naming URL in the naming request.

Property: `com.sun.identity.agents.config.ignore.preferred.naming.url`

Invalid URL properties (Not yet in OpenAM console)

Invalid URL Regular Expression

Use a Perl-compatible regular expression to filter out invalid request URLs. The policy agent rejects requests to invalid URLs with HTTP 403 Forbidden status without further processing. Use Advanced > Custom Properties to set this in the agent profile.

For example, to filter out URLs containing the symbols in the list ./, /., /, ., ,\, %00-%1f, %7f-%ff, %25, %2B, %2C, %7E, .info, use the following setting.

```
com.forgerock.agents.agent.invalid.url.regex= \
     ^((?!(|/\.|\./||*|\.info|%25|%2B|%2C|%[0-1][0-9a-fA-F]|%[7-9a-fA-F][0-9a-fA-F])).)$
```

Ignore Server Check Properties

Ignore Server Check

When enabled, do not check whether OpenAM is up before doing a 302 redirect.

Property: `com.sun.identity.agents.config.ignore.server.check`

Ignore Path Info Properties

Ignore Path Info in Request URL

When enabled, strip path info from the request URL while doing the Not Enforced List check, and URL policy evaluation. This is designed to prevent a user from accessing a URI by appending the matching pattern in the policy or not enforced list.

For example, if the not enforced list includes `http://host/*.gif`, then stripping path info from the request URI prevents access to `http://host/index.html` by using `http://host/index.html?hack.gif`.

However, when a web server is configured as a reverse proxy for a J2EE application server, the path info is interpreted to map a resource on the proxy server rather than the application server. This prevents the not enforced list or the policy from being applied to the part of the URI below the application server path if a wildcard character is used.

For example, if the not enforced list includes `http://host/webapp/servcontext/*` and the request URL is `http://host/webapp/servcontext/example.jsp`, the path info is `/servcontext/example.jsp` and the resulting request URL with path info stripped is `http://host/webapp/`, which does not match the not enforced list. Thus when this property is enabled, path info is not stripped from the request URL even if there is a wildcard in the not enforced list or policy.

Make sure therefore when this property is enabled that there is nothing following the wildcard in the not enforced list or policy.

Property: `com.sun.identity.agents.config.ignore.path.info`

Multi-Byte Enable Properties

Native Encoding of Profile Attributes

When enabled, the agent encodes the LDAP header values in the default encoding of operating system locale. When disabled, the agent uses UTF-8.

Property: `com.sun.identity.agents.config.convert.mbyte.enable`

Goto Parameter Name Properties

Goto Parameter Name

Property used only when CDSSO is enabled. Only change the default `goto` value when the login URL has a landing page specified, such as `com.sun.identity.agents.config.cdsso.cdcservlet.url = http://openam.example.com:8080/openam/cdcservlet?goto= http://www.example.com/landing.jsp`. The agent uses this parameter to append the original request URL to this cdcservlet URL. The landing page consumes this parameter to redirect to the original URL.

As an example, if you set this value to `goto2`, then the complete URL sent for authentication is `http://openam.example.com:8080/openam/cdcservlet?goto=`

`http://www.example.com/landing.jsp?goto2=http://www.example.com/original.jsp`.

Property: `com.sun.identity.agents.config.redirect.param`

Deprecated Agent Properties

Anonymous User Default Value
User ID of unauthenticated users. Default: `anonymous`.

Property: `com.sun.identity.agents.config.anonymous.user.id`

3.6.6 Configuring Web Policy Agent Advanced Properties

This section covers advanced web agent properties. After creating the agent profile, you access these properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* > Advanced.

This section describes the following property groups:

- Client Identification Properties
- Load Balancer Properties
- Post Data Preservation Properties
- Sun Java System Proxy Server Properties
- Microsoft IIS Server Properties
- IBM Lotus Domino Server Properties
- Custom Properties

Client Identification Properties

If the agent is behind a proxy or load balancer, then the agent can get client IP and host name values from the proxy or load balancer. For proxies and load balancer that support providing the client IP and host name in HTTP headers, you can use the following properties.

When multiple proxies or load balancers sit in the request path, the header values can include a comma-separated list of values with the first value representing the client, as in `client,next-proxy,first-proxy`.

Client IP Address Header
HTTP header name that holds the IP address of the client.

Property: `com.sun.identity.agents.config.client.ip.header`

Client Hostname Header

HTTP header name that holds the hostname of the client.

Property: `com.sun.identity.agents.config.client.hostname.header`

Load Balancer Properties

Load Balancer Setup

Enable if a load balancer is used for OpenAM services.

Property: `com.sun.identity.agents.config.load.balancer.enable`

Hot swap: no

Override Request URL Protocol

Enable if the agent is sitting behind a SSL/TLS off-loader, load balancer, or proxy such that the protocol users use is different from the protocol the agent uses. When enabled, the protocol is overridden with the value from the Agent Deployment URI Prefix (property: `com.sun.identity.agents.config.agenturi.prefix`).

Property: `com.sun.identity.agents.config.override.protocol`

Override Request URL Host

Enable if the agent is sitting behind a SSL/TLS off-loader, load balancer, or proxy such that the host name users use is different from the host name the agent uses. When enabled, the host is overridden with the value from the Agent Deployment URI Prefix (property: `com.sun.identity.agents.config.agenturi.prefix`).

Property: `com.sun.identity.agents.config.override.host`

Override Request URL Port

Enable if the agent is sitting behind a SSL/TLS off-loader, load balancer, or proxy such that the port users use is different from the port the agent uses. When enabled, the port is overridden with the value from the Agent Deployment URI Prefix (property: `com.sun.identity.agents.config.agenturi.prefix`).

Property: `com.sun.identity.agents.config.override.port`

Override Notification URL

Enable if the agent is sitting behind a SSL/TLS off-loader, load balancer, or proxy such that the URL users use is different from the URL the agent uses. When enabled, the URL is overridden with the value from the Agent Deployment URI Prefix (property: `com.sun.identity.agents.config.agenturi.prefix`).

Property: `com.sun.identity.agents.config.override.notification.url` `com.sun.identity.agents.config.postdata.preserve.stickysession.mode` (Not yet in OpenAM Console)

Specifies whether to create a cookie, or to append a query string to the URL to assist with sticky load balancing.

`com.sun.identity.agents.config.postdata.preserve.stickysession.value` (Not yet in OpenAM Console)

Specifies the key-value pair for stickysession mode. For example, a setting of `lb=myserver` either sets an `lb` cookie with `myserver` value, or adds `lb=myserver` to the URL query string.

Post Data Preservation Properties

POST Data Preservation

Enables HTTP POST data preservation. This feature is available in the Apache 2.2, Microsoft IIS 6, Microsoft IIS 7, and Sun Java System Web Server web policy agents as of version 3.0.3.

Property: `com.sun.identity.agents.config.postdata.preserve.enable`

POST Data Entries Cache Period

POST cache entry lifetime in minutes. Default: 10.

Property: `com.sun.identity.agents.config.postcache.entry.lifetime`

POST Data Preservation Cookie Name (Not yet in OpenAM Console)

When HTTP POST data preservation is enabled, override properties are set to true, and the agent is behind a load balancer, then this property sets the name and value of the sticky cookie to use.

Property: `com.sun.identity.agents.config.postdata.preserve.lbcookie`

`org.forgerock.agents.config.postdata.preserve.dir` (Not yet in OpenAM Console)

The directory on the agent server where preserved post data will be written whilst authorization is requested from OpenAM.

Default: `/web_agents/`*`agent_version`*`/log`

Post Data Preservation URI Prefix (Not yet in OpenAM Console)

If you run multiple web servers with policy agents behind a load balancer that directs traffic based on the request URI, and you need to preserve POST data, then set this property.

By default, policy agents use a dummy URL for POST data preservation, `http://`*`agent.host`*`:`*`port`*`/dummypost/sunpostpreserve`, to handle POST data across redirects to and from OpenAM. When you set this property, the policy agent prefixes the property value to the dummy URL path. In other words, when you set `com.forgerock.agents.config.pdpuri.prefix = app1`, the policy agent uses the dummy URL, `http://agent.host:port/app1/dummypost/sunpostpreserve`.

Next, use the prefix you set when you define load balancer URI rules. This ensures that clients end up being redirected to the policy agent that preserved the POST data.

Property: `com.forgerock.agents.config.pdpuri.prefix`

`org.forgerock.agents.pdp.javascript.repost` (Not yet in OpenAM Console)
When set to `true`, preserved post data will be resubmitted to the destination server after authentication by using JavaScript.

Sun Java System Proxy Server Properties

Override Proxy Server's Host and Port
When enabled ignore the host and port settings.

Property: `com.sun.identity.agents.config.proxy.override.host.port`

Hot swap: no

Microsoft IIS Server Properties

Authentication Type
The agent should normally perform authentication, so this is not required. If necessary, set to `none`.

Property: `com.sun.identity.agents.config.iis.auth.type`

Hot swap: no

Replay Password Key
DES key for decrypting the basic authentication password in the session.

Property: `com.sun.identity.agents.config.replaypasswd.key`

Filter Priority
The loading priority of filter, DEFAULT, HIGH, LOW, or MEDIUM.

Property: `com.sun.identity.agents.config.iis.filter.priority`

Filter configured with OWA
Enable if the IIS agent filter is configured for OWA.

Property: `com.sun.identity.agents.config.iis.owa.enable`

Change URL Protocol to HTTPS
Enable to avoid IE6 security pop-ups.

Property: `com.sun.identity.agents.config.iis.owa.enable.change.protocol`

Idle Session Timeout Page URL

This property is no longer used.

Property: `com.sun.identity.agents.config.iis.owa.enable.session.timeout.url`

Show Password in HTTP Header

Set to `true` if encrypted password should be set in HTTP header `AUTH_PASSWORD`.

Property: `com.sun.identity.agents.config.iis.password.header`

Logon and Impersonation

Set to `true` if agent should do Windows Logon and User Impersonation.

Property: `com.sun.identity.agents.config.iis.logonuser`

IBM Lotus Domino Server Properties

Check User in Domino Database

When enabled, the agent checks whether the user exists in the Domino name database.

Property: `com.sun.identity.agents.config.domino.check.name.database`

Use LTPA token

Enable if the agent needs to use LTPA Token.

Property: `com.sun.identity.agents.config.domino.ltpa.enable`

LTPA Token Cookie Name

The name of the cookie that contains the LTPA token.

Property: `com.sun.identity.agents.config.domino.ltpa.cookie.name`

LTPA Token Configuration Name

The configuration name that the agent uses in order to employ the LTPA token mechanism.

Property: `com.sun.identity.agents.config.domino.ltpa.config.name`

LTPA Token Organization Name

The organization name to which the LTPA token belongs.

Property: `com.sun.identity.agents.config.domino.ltpa.org.name`

Custom Properties

Custom Properties

Additional properties to augment the set of properties supported by agentd. Such properties take the following forms.

- `customproperty=custom-value1`
- `customlist[0]=customlist-value-0`
- `customlist[1]=customlist-value-1`
- `custommap[key1]=custommap-value-1`
- `custommap[key2]=custommap-value-2`

Property: `com.sun.identity.agents.config.freeformproperties`

3.6.7 Configuring Web Policy Agent Custom Properties

This section covers custom web agent properties.

Note

These settings do not appear as configurable options in the OpenAM Console, so must be added as custom properties, or set in the local configuration file.

If using a centralized configuration, you create these properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* > Advanced > Custom Properties.

This section describes the following property groups:

- Bootstrap Properties
- Encryption Properties
- Miscellaneous Custom Properties

Bootstrap Properties

These properties are only used within the local configuration file. They are not available in the OpenAM admin console. The agent uses these bootstrap properties to connect to OpenAM.

`com.sun.identity.agents.config.organization.name`

The OpenAM realm where the agent profile is located.

Default: `/`

`com.sun.identity.agents.config.username`
The name of the agent profile in OpenAM.

`com.sun.identity.agents.config.password`
The password required by the agent profile, encrypted with the key specified in `com.sun.identity.agents.config.key`.

`com.sun.identity.agents.config.key`
The encryption key used to encrypt the agent profile password, which should be provided in `com.sun.identity.agents.config.password`.

`org.forgerock.agents.config.tls`
Set this property to a list of protocols to support. The list consists of one or more protocol strings separated by colons. For example `TLSv1.2:TLSv1.1:TLSv1`.

Encryption Properties

`com.forgerock.agents.config.cert.ca.file`
Set this property to the file name that contains one or more CA certificates. The file should be *Privacy Enhanced Mail* (PEM) encoded. OpenAM requires PEM files to be base64-encoded ASCII data.

You must set this property if `com.sun.identity.agents.config.trust.server.certs` is set to `false`.

`com.forgerock.agents.config.cert.file`
When OpenAM is configured to perform client authentication, set this property to the name of the file that contains the public PEM-encoded client certificate that corresponds with the private key specified in `com.forgerock.agents.config.cert.key`.

`com.forgerock.agents.config.cert.key`
Set this property to the name of the file that contains the private key. On UNIX systems, that key should be encoded in PEM format.

On Windows systems, that entry depends. If SSL mutual authentication is required with OpenAM, that entry should contain the name of the private key or certificate imported in the Windows Certificate Manager, part of the Microsoft Management Console. For a web server, that should point to the Local Machine or Service certificate store, depending on the account associated with the Web server.

`com.forgerock.agents.config.cert.key.password`
Set this property to the obfuscated private key password. Obfuscate the password by using **agentadmin --p**, as demonstrated in the following example to generate the value:

```
$ cd /web_agents/agent-type/bin
$ ./agentadmin --p "key" "password"
```

Here, *agent-type* corresponds to the file system directory for the particular agent type, such as apache24_agent, *password* is the private key password, and *key* is the obfuscation key as specified by com.sun.identity.agents.config.key.

Tip

You can generate a new obfuscation key by using **agentadmin --k**.

This property is not used on Microsoft Windows systems.

com.forgerock.agents.config.ciphers
Set this property to a list of ciphers to support. The list consists of one or more cipher strings separated by colons, as defined in the man page for ciphers available at http://www.openssl.org/docs/apps/ciphers.html.

Default: HIGH:MEDIUM.

com.sun.identity.agents.config.trust.server.certs
When SSL is configured, set to false to trust the OpenAM SSL certificate only if the certificate is found to be correct and valid. Default is true to make it easy to try SSL during evaluation.

Important

Notice that the default setting, true, means that the web policy agent trusts all server certificates. Change this to false, and test that your web policy agent can trust server certificates before deploying the policy agent in production.

Naming URL and Failover Properties

com.forgerock.agents.ext.url.validation.default.url.set
This property takes a comma-separated list of indexes for URL values indicating the order in which to fail over, where the indexes are taken from the values set for com.sun.identity.agents.config.naming.url, com.sun.

identity.agents.config.login.url, com.sun.identity.agents.config.cdsso.cdcservlet.url, and com.sun.identity.agents.config.logout.url.

For example if com.sun.identity.agents.config.naming.url is set as follows:

```
com.sun.identity.agents.config.naming.url=
 http://zero.example.com:8080/openam/namingservice
 http://one.example.com:8080/openam/namingservice
```

Then the following setting means first use OpenAM on zero.example.com, then fail over if necessary to OpenAM on one.example.com, assuming com.forgerock.agents.ext.url.validation.level is set to enable validation.

```
com.forgerock.agents.ext.url.validation.default.url.set=0,1
```

When using this failover capability make sure you synchronize URL settings in com.sun.identity.agents.config.naming.url, com.sun.identity.agents.config.login.url, com.sun.identity.agents.config.cdsso.cdcservlet.url, and com.sun.identity.agents.config.logout.url such that each service shares the same index across all properties. In other words, in the example above each service under http://zero.example.com:8080/openam would be the first item (index: 0) for each property. This ensures the policy agent fails over and fails back from one server to another in synchronized fashion for all services.

This property has no default setting.

com.forgerock.agents.ext.url.validation.level

This bootstrap configuration property lets you configure naming URL validation during the initial bootstrap phase when the policy agent reads its configuration, and then thereafter if the policy agent is configured fail over when a naming URL becomes invalid.

When URL validation is fully disabled the policy agent does not need to connect to OpenAM during the bootstrap phase.

If you leave naming URL validation disabled, then make sure that the URLs in the policy agent bootstrap configuration file are valid and correct. As the policy agent performs no further validation after the bootstrap phase, incorrect naming URLs can cause the agent to crash.

To enable full URL validation, set the property as shown:

```
com.forgerock.agents.ext.url.validation.level = 0
```

This property can take the following values.

0

Fully validate naming URLs specified by using the com.sun.identity.agents.config.naming.url property. The web policy agent logs into and logs out of OpenAM to check that a naming URL is valid.

1
Check that naming URLs are valid by performing an HTTP GET, which should receive an HTTP 200 response.

2 (Default)
Disable all naming URL validation.

When naming URL validation is enabled, then set the following properties.

- `com.sun.identity.agents.config.connect.timeout`
- `com.sun.identity.agents.config.receive.timeout`

`com.forgerock.agents.ext.url.validation.ping.interval`
Set this to the seconds between validation requests against the current naming URL.

The sum of the values of `com.sun.identity.agents.config.connect.timeout` and `com.sun.identity.agents.config.receive.timeout` must not exceed this value. Notice that the two timeout values are specified in milliseconds, whereas this property's value is specified in seconds.

Default: 60 (seconds)

`com.sun.identity.agents.config.connect.timeout`
Set this to the number of milliseconds to keep the socket connection open before timing out. If you have the web policy agent perform naming URL validation, then set this property to a reasonable value such as 2000 (2 seconds). The default value is 0 which implies no timeout.

`com.forgerock.agents.ext.url.validation.ping.miss.count`
If validation requests against the current naming URL fail this number of times in a row, the web policy agent fails over to the next service in `com.forgerock.agents.ext.url.validation.default.url.set`.

Default: 3

`com.forgerock.agents.ext.url.validation.ping.ok.count`
After failover, if validation requests against the default naming URL succeed this number of times in a row, the web policy agent fails back to that service, the first URL in the `com.forgerock.agents.ext.url.validation.default.url.set` list.

Default: 3

`com.sun.identity.agents.config.naming.url`
Set this to the naming service URL(s) used for naming lookups in OpenAM. Separate multiple URLs with single space characters.

Miscellaneous Custom Properties

`com.forgerock.agents.cache_control_header.enable`
Set this property to `true` to enable use of Cache-Control headers that prevent proxies from caching resources accessed by unauthenticated users. Default: `false`.

`org.forgerock.agents.config.json.url`
Use regular expressions to specify a list of resource URLs that should trigger JSON-formatted errors to be returned rather than HTTP error codes.

`org.forgerock.agents.config.keepalive.disable`
The web policy agents by default use a single connection and specify `Connection:Keep-alive` when logging in to OpenAM and fetching attributes or policy decisions.

If a load-balancer or reverse-proxy is being used it may be necessary to disable the use of keep-alive, in which case set this property to `true`.

Default: false

3.6.8 Configuring Web Policy Agent Environment Variables

This section covers web agent properties that are configured by using environment variables. You must restart the container in which web policy agents are running to apply changes to these settings.

Web Policy Agent Environment Properties

`AM_MAX_SHARED_POOL_SIZE`
Configure the maximum amount of shared memory, in bytes, that the web policy agents use for caching. The maximum size the cache can grow to is approximately 2 gigabytes (exactly 0x7FFFF000 bytes).

You can reduce the maximum size by setting `AM_MAX_SHARED_POOL_SIZE`, specified in bytes. You should not reduce the cache size to less than 10 megabytes. You cannot increase the default maximum cache size.

Warning

Reducing the size of the cache may affect web policy agent performance under heavy workloads, such as handling thousands of concurrent sessions.

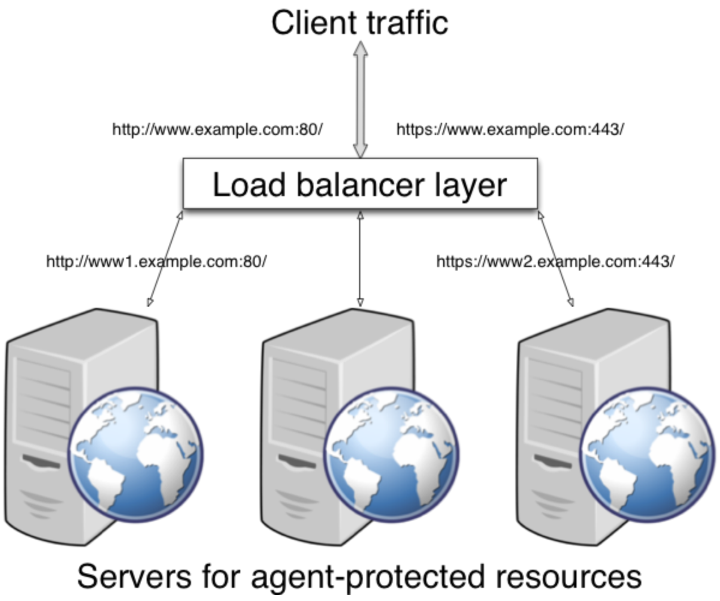
3.7 Configuring Web Policy Agents Behind Load Balancers

This chapter addresses the question of configuring policy agents on protected servers that operate behind network load balancers.

3.7.1 The Role of the Load Balancing Layer

A load balancing layer that stands between clients and protected servers can distribute the client load, and fail client traffic over when a protected server goes offline. In the simplest case, the load balancing layer passes requests from the clients to servers and responses from servers to clients, managing the traffic so the client experience is as smooth as possible.

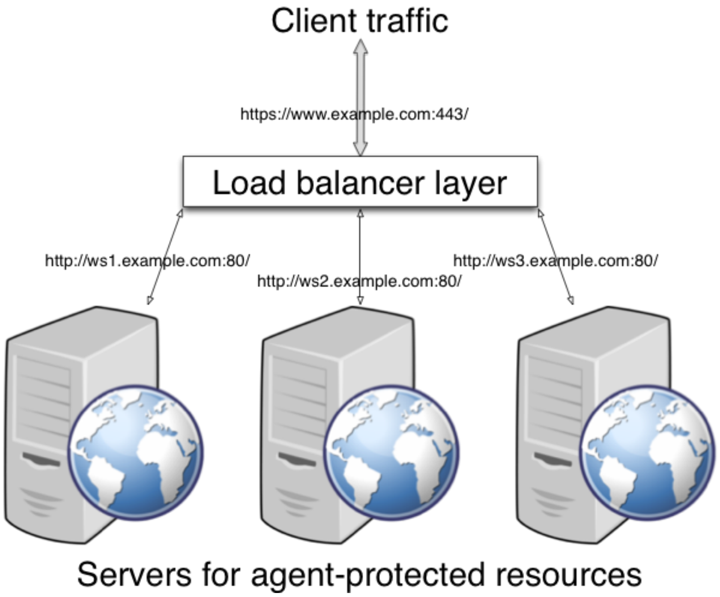
Figure 3.1. Load Balancing With the Same Protocol and Port


Client traffic
http://www.example.com:80/
https://www.example.com:443/
Load balancer layer
http://www1.example.com:80/
https://www2.example.com:443/
Servers for agent-protected resources

If your deployment has protocols and port numbers on the load balancer that match those of the protected servers, see Section 3.7.2, "When Protocols and Port Number Match".

A load balancing layer can also offload processor-intensive public-key encryption algorithms involved in SSL transactions to a hardware accelerator, reducing the load on the protected servers. The client connects to the load balancer over HTTPS, but the load balancer connects to the servers over HTTP.

Figure 3.2. Load Balancing With SSL Offloading


Client traffic
https://www.example.com:443/
Load balancer layer
http://ws1.example.com:80/
http://ws2.example.com:80/
http://ws3.example.com:80/
Servers for agent-protected resources

If your deployment uses SSL offloading, see Section 3.7.3, "When Protocols and Port Number Differ".

3.7.2 When Protocols and Port Number Match

When the protocol on the load balancer, such as HTTP or HTTPS, matches the protocol on the protected web server, and the port number the load balancer listens on, such as 80 or 443, matches the port number the protected web server listens on, then the main difference between URLs is in the host names. Map the agent host name to the host name for the load balancer.

Procedure 3.5. To Map the Agent Host Name to the Load Balancer Host Name

When protocols and port numbers match, configure fully qualified domain name (FQDN) mapping.

This procedure explains how to do so for a centralized web policy agent profile configured in OpenAM Console. The steps also mention the properties for web agent profiles that rely on local, file-based configurations:

1. Login to OpenAM Console as an administrative user with rights to modify the policy agent profile.

2. Browse to Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* to open the web agent profile for editing.

3. In the Global tab page section Fully Qualified Domain Name Checking, make sure FQDN checking is selected (the default).

   The equivalent property setting is `com.sun.identity.agents.config.fqdn.check.enable=true`.

4. Set FQDN Default to the fully qualified domain name of the load balancer, such as `lb.example.com`, rather than the protected server FQDN where the policy agent is installed.

   The equivalent property setting is `com.sun.identity.agents.config.fqdn.default=lb.example.com`.

5. Set FQDN Virtual Host Map to map the protected server FQDN to the load balancer FQDN, for example, where the key `agent.example.com` (protected server) has value `lb.example.com` (load balancer).

   The equivalent property setting is `com.sun.identity.agents.config.fqdn.mapping[agent.example.com]=lb.example.com`.

6. Save your work, and then restart the protected server.

3.7.3 When Protocols and Port Number Differ

When the load balancer protocol and port, such as HTTPS and 443, differ from the protocol on the protected web server, such as HTTP and 80, then you must override these in the policy agent configuration.

Procedure 3.6. To Override Protocol, Host, and Port

Use the Agent Deployment URI Prefix setting to override the agent protocol, host, and port with that of the load balancer.

Important

The web policy agent configuration for SSL offloading has the side effect of preventing FQDN checking and mapping. As a result, URL rewriting and redirection does not work correctly when the policy agent is accessed directly and not through the load balancer. This should not be a problem for client traffic, but potentially could be an issue for applications accessing the protected server directly, from behind the load balancer.

This procedure explains how to do so for a centralized web policy agent profile configured in OpenAM Console. The steps also mention the properties for web agent profiles that rely on local, file-based configurations:

1. Login to OpenAM Console as an administrative user with rights to modify the policy agent profile.

2. Browse to Realms > *`Realm Name`* > Agents > Web > *`Agent Name`* to open the web agent profile for editing.

3. In the Global tab page Profile section, set the Agent Deployment URI Prefix to that of the load balancer.

   The value you set here is used when overriding protocol, host, and port on the protected server with the web policy agent.

   The property to set is `com.sun.identity.agents.config.agenturi.prefix`.

4. In the Advanced tab page Load Balancer section, enable Load Balancer Setup.

   The equivalent property setting is `com.sun.identity.agents.config.load.balancer.enable=true`.

5. Enable Override Request URL Protocol.

   The equivalent property setting is `com.sun.identity.agents.config.override.protocol=true`.

6. Enable Override Request URL Host.

   The equivalent property setting is `com.sun.identity.agents.config.override.host=true`.

7. Enable Override Request URL Port.

   The equivalent property setting is `com.sun.identity.agents.config.override.port=true`.

8. Enable Notification URL when the web policy agent gets notifications about configuration changes.

   The equivalent property setting is `com.sun.identity.agents.config.override.notification.url=true`.

9. Save your work, and then restart the protected server.

3.8 Configuring Agent Authenticators

An *agent authenticator* has read-only access to multiple agent profiles defined in the same realm, typically allowing an agent to read web service agent profiles.

After creating the agent profile, you access agent properties in the OpenAM console under Realms > *`Realm Name`* > Agents > Agent Authenticator > *`Agent Name`*.

Password
: Specifies the password the agent uses to connect to OpenAM.

Status
: Specifies whether the agent profile is active, and so can be used.

Agent Profiles allow to Read
: Specifies which agent profiles in the realm the agent authenticator can read.

Agent Root URL for CDSSO
: Specifies the list of agent root URLs for CDSSO. The valid value is in the format *`protocol`*`://`*`hostname`*`:`*`port`*`/` where *`protocol`* represents the protocol used, such as `http` or `https`, *`hostname`* represents the host name of the system where the agent resides, and *`port`* represents the port number on which the agent is installed. The slash following the port number is required.

  If your agent system also has virtual host names, add URLs with the virtual host names to this list as well. OpenAM checks that `goto` URLs match one of the agent root URLs for CDSSO.

Chapter 4

Installing Web Policy Agents in Apache HTTP Server

This chapter covers prerequisites and installation procedures for Web Policy Agents 4 into Apache HTTP Servers 2.2.x and 2.4.x.

4.1 Before You Install

This section describes the prerequisite steps you should take before installing the web policy agents into Apache HTTP servers.

- Avoid installing the web server and the web policy agent as root. Instead, create a web server user and install as that user.

  If you cannot avoid installing the web server and web policy agent as root, then you must give all users read and write permissions to the `logs` and `logs/debug` directories under the agent instance directory (`/web_agents/`*`type`*`/Agent_`*`nnn`*`/logs/`). Otherwise, the web policy agent fails with an error when attempting to rotate log files.

  Tip

  The installer can automatically set permissions on folders that require write access, by reading the Apache config file to determine the correct group and user to grant privileges to. Answer yes when prompted:

```
Change ownership of created directories using
User and Group settings in httpd.conf
[ q or 'ctrl+c' to exit ]
(yes/no): [no]: yes
```

- The *SELinux* OS feature can prevent the agents from being able to write to audit and debug logs. See Chapter 6, "Troubleshooting".

- Ensure OpenAM is installed and running, so that you can contact OpenAM from the system running the policy agent.

- Create a profile for your policy agent as described in Chapter 3, "Configuring Web Policy Agents".

- Create at least one policy in OpenAM to protect resources with the agent, as described in the section on *Configuring Policies*. Consider creating a simple policy, such as a policy that allows only authenticated users to access your resources. This allows you to test your policy agent after installation.

- If the OpenAM server uses SSL, you must install OpenSSL on the agent machine.

  - On UNIX systems, ensure the OpenSSL libraries `libcrypto.so` and `libssl.so` are available in the path specified by either the `LD_LIBRARY_PATH` or `LD_LIBRARY_PATH_64` environment variables.

  - On Windows systems, ensure the OpenSSL libraries `libeay32.dll` and `ssleay32.dll` are available in the `lib` folder of your agent installation, for example `c:\path\to\web_agents\iis_agent\lib\`.

- Install Apache HTTP Server before you install the policy agent. You must stop the server during installation.

- See the OpenAM *Installation Guide* section, *Obtaining OpenAM Software* to determine which version of the agent to download, and download the agent. Also, verify the checksum of the file you download against the checksum posted on the download page.

  Unzip the file in the directory where you plan to install the web policy agent. The agent stores its configuration and logs under this directory.

  When you unzip the policy agent .zip download, you find the following directories:

`bin`
: The installation and configuration program **agentadmin**.

`config`
: Configuration templates used by the **agentadmin** command during installation.

`instances`
: Configuration files, and audit and debug logs for individual instances of the web policy agents will be created here. The folder is empty when first extracted.

`legal`
: Contains licensing information including third-party licenses.

`lib`
: Shared libraries used by the policy agent.

`log`
: Location for log files written during installation. The folder is empty when first extracted.

4.1.1 Tuning Apache Multi-Processing Modules

Apache 2.0 and later comes with Multi-Processing Modules (MPMs) that extend the basic functionality of a web server to support the wide variety of operating systems and customizations for a particular site.

The key area of performance tuning for Apache is to run in worker mode ensuring that there are enough processes and threads available to service the expected number of client requests. Apache performance is configured in the `conf/extra/http-mpm.conf` file.

The key properties in this file are `ThreadsPerChild` and `MaxClients`. Together the properties control the maximum number of concurrent requests that can be processed by Apache. The default configuration allows for 150 concurrent clients spread across 6 processes of 25 threads each.

```
<IfModule mpm_worker_module>
   StartServers          2
   MaxClients          150
   MinSpareThreads      25
   MaxSpareThreads      75
   ThreadsPerChild      25
   MaxRequestsPerChild   0
</IfModule>
```

Important

For the policy agent notification feature, the `MaxSpareThreads`, `ThreadLimit` and `ThreadsPerChild` default values must *not* be altered; otherwise the notification queue listener thread cannot be registered.

Any other values apart from these three in the worker MPM can be customized. For example, it is possible to use a combination of `MaxClients` and `ServerLimit` to achieve a high level of concurrent clients.

4.2 Installing Apache Web Policy Agents

Complete the following procedures to install Web Policy Agents 4 into Apache HTTP Servers.

Tip

Check that you have completed any prerequisite steps before proceeding. See Section 4.1, "Before You Install".

There are two web policy agents packages available for Apache installs:

Apache 2.2
: Available in 32-bit and 64-bit. By default, extracts to a folder named `./web_agents/apache22_agent/`.

Apache 2.4
: Available in 32-bit and 64-bit. By default, extracts to a folder named `./web_agents/apache24_agent/`.

Tip

The following procedures show how to install into Apache 2.4. If installing into Apache 2.2, alter the path names accordingly.

Procedure 4.1. To Create the Agent Profile

Regardless of whether you store configurations centrally in OpenAM or locally with your agents, the agent requires a profile so that it can connect to and communicate with OpenAM.

1. In the OpenAM console, browse to Realms > *`Realm Name`* > Agents > Web, and then click the New... button in the Agent table.

2. Complete the web form using the following hints.

   Name
   : The name for the agent profile used when you install the agent

   Password
   : Password the agent uses to authenticate to OpenAM

   Configuration
   : Centralized configurations are stored in the OpenAM configuration store. You can manage the centralized configuration through the OpenAM console. Local configurations are stored in a file alongside the agent.

   Server URL
   : The full URL to an OpenAM instance, or if OpenAM is deployed in a site configuration (behind a load balancer) then the site URL

     In centralized configuration mode, the Server URL is used to populate the agent profile for services, such as Login, Logout, Naming, and Cross Domain SSO.

   Agent URL
   : The URL to the web agent application, such as `http://www.example.com:80`

     In centralized configuration mode, the Agent URL is used to populate the Agent Profile for services, such as notifications.

Procedure 4.2. To Create a Password File

1. Create a text file containing only the password specified when creating the agent profile.

   UNIX example:

   ```
   $ echo password > /tmp/pwd.txt
   ```

   Windows example:

```
C:\> echo password > pwd.txt
```

2. Protect the password file you create as appropriate for your operating system:

   UNIX example:

   ```
   $ chmod 400 /tmp/pwd.txt
   ```

   Windows example:

   In Windows Explorer, right-click the created password file, for example `pwd.txt`, select Read-Only, and then click OK.

Procedure 4.3. To Install the Web Policy Agent into Apache

1. Shut down the Apache server where you plan to install the agent.

2. Make sure OpenAM is running.

3. Run **agentadmin --i** to install the agent. You will be prompted to read and accept the software license agreement for the agent installation.

   - UNIX example:

     ```
     $ cd /web_agents/apache24_agent/bin/
     $ ./agentadmin --i
     ```

   - Windows example:

     ```
     C:\> cd web_agents\apache24_agent\bin
     C:\path\to\web_agents\apache24_agent\bin> agentadmin.exe --i
     ```

4. When prompted for information, enter the inputs appropriate for your deployment.

   ### Tip

   You can cancel web policy agent installation at anytime by pressing **CTRL+C**

   a. Enter the full path to the Apache HTTP Server configuration file. The installer modifies this file to include the web policy agent configuration and module.

```
Enter the complete path to the httpd.conf file which is used by Apache HTTPD
Server to store its configuration.
[ q or 'ctrl+c' to exit ]
Configuration file [/opt/apache/conf/httpd.conf]: /etc/httpd/conf/httpd.conf
```

b. The installer can change the directory ownership to the same User and Group specified in the Apache configuration. Enter `yes` to alter directory ownership, press **Enter** to accept the default: `no`.

```
Change ownership of created directories using
User and Group settings in httpd.conf
[ q or 'ctrl+c' to exit ]
(yes/no): [no]: yes
```

c. The installer can import settings from an existing web policy agent into the new installation and skips prompts for any values present in the existing configuration file. You will be required to re-enter the agent profile password.

   Enter the full path to an existing agent configuration file to import the settings, or press **Enter** to skip the import.

```
To set properties from an existing configuration enter path to file
[ q or 'ctrl+c' to exit, return to ignore ]
Existing agent.conf file:
```

d. Enter the full URL of the OpenAM instance the web policy agents will be using. Ensure that the deployment URI is specified.

```
Enter the URL where the OpenAM server is running. Please include the
deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ q or 'ctrl+c' to exit ]
OpenAM server URL: http://openam.example.com:8080/openam
```

e. Enter the full URL of the server the agent is running on.

```
Enter the Agent URL as shown below:
(http://agent.sample.com:1234)
[ q or 'ctrl+c' to exit ]
Agent URL: http://www.example.com:80
```

f. Enter the name given to the agent profile created in OpenAM.

```
Enter the Agent profile name
[ q or 'ctrl+c' to exit ]
Agent Profile name: webagent4
```

g. Enter the OpenAM realm containing the agent profile.

```
Enter the Agent realm/organization
[ q or 'ctrl+c' to exit ]
Agent realm/organization name: [/]: /
```

h. Enter the full path to the file containing the agent profile password created earlier.

```
Enter the path to a file that contains the password to be used
for identifying the Agent
[ q or 'ctrl+c' to exit ]
The path to the password file: /tmp/pwd.txt
```

i. The installer displays a summary of the configuration settings you specified.

- If a setting is incorrect, type `no`, or press **Enter**. The installer loops through the configuration prompts again, using your provided settings as the default. Press **Enter** to accept each one, or enter a replacement setting.

- If the settings are correct, type `yes` to proceed with installation.

```
Installation parameters:

   OpenAM URL: http://openam.example.com:8080/openam
   Agent URL: http://www.example.com:80
   Agent Profile name: webagent4
   Agent realm/organization name: /
   Agent Profile password source: /tmp/pwd.txt

Confirm configuration (yes/no): [no]: yes
Validating...
Validating... Success.
Cleaning up validation data...
Creating configuration...
Installation complete.
```

Upon successful completion, the installer adds the agent as a module to the Apache HTTP Server configuration file. You can find a backup configuration file in the Apache HTTP Server configuration directory, called `http.conf_amagent_`*`date_and_time_of_installation`*.

The installer also sets up configuration and log directories for the agent instance. Each agent instance that you install on the system has its own numbered configuration and logs directory. The first agent's configuration and logs are located under the directory `web_agents/apache24_agent/instances/agent_1/`.

The configuration files and log locations are as follows:

`config/agent.conf`
: Contains the bootstrap properties the web policy agent requires to connect to OpenAM and download its configuration. Also contains properties that are only used if you configure the web policy agent to use local configuration.

`logs/audit/`
: Operational audit log directory, only used if remote logging to OpenAM is disabled.

`logs/debug/`
: Debug directory where the `amAgent` debug file resides. Useful in troubleshooting policy agent issues.

5. Start the Apache server in which you installed the web policy agent.

Procedure 4.4. To Check the Policy Agent Installation

1. Check the Apache HTTP server error log after you start the server to make sure startup completed successfully:

```
[Tue Sep 08 15:51:27.667625 2015] AH00163:
 Apache/2.4.6 (CentOS) OpenAM Web Agent/4 configured
 -- resuming normal operations
```

2. Check the `/web_agents/apache24_agent/instances/Agent_1/logs/debug/debug.log` file to verify that no errors occurred on startup. Expected output should resemble the following:

```
2015-09-08 16:02:24.573 -0700 INFO [0x7f7470064840:5748]

######################################################
  OpenAM Web Agent
  Version: 4
  Revision: 15441
  Build date: Aug 29 2015 02:48:01
######################################################
```

3. If you have a policy configured, you can test your policy agent. For example, try to browse to a resource that your policy agent protects. You should be redirected to OpenAM to authenticate, for example, as user `demo`, password *`changeit`*. After you authenticate, OpenAM redirects you back to the resource you tried to access.

4.3 Installing Apache Web Policy Agents into a Virtual Host

Complete the following procedures to install Web Policy Agents 4 into Apache HTTP Server virtual hosts.

Installing into an Apache virtual host is a manual process, which involves copying an instance directory created by the **agentadmin** installer and adding to the Apache configuration file of the virtual host.

Tip

Check that you have completed the prerequisite steps before proceeding. See Section 4.1, "Before You Install".

You will also need to have installed a web policy agent into the default root Apache configuration file before installing into a virtual host. See Section 4.2, "Installing Apache Web Policy Agents".

Procedure 4.5. To Create the Agent Profile

Regardless of whether you store configurations centrally in OpenAM or locally with your agents, the agent requires a profile so that it can connect to and communicate with OpenAM.

1. In the OpenAM console, browse to Realms > *`Realm Name`* > Agents > Web, and then click the New... button in the Agent table.

2. Complete the web form using the following hints.

   Name
   : The name for the agent profile used when you install the agent

   Password
   : Password the agent uses to authenticate to OpenAM

   Configuration
   : Centralized configurations are stored in the OpenAM configuration store. You can manage the centralized configuration through the OpenAM console. Local configurations are stored in a file alongside the agent.

   Server URL
   : The full URL to an OpenAM instance, or if OpenAM is deployed in a site configuration (behind a load balancer) then the site URL

In centralized configuration mode, the Server URL is used to populate the agent profile for services, such as Login, Logout, Naming, and Cross Domain SSO.

Agent URL

The URL to the web agent application, such as `http://www.example.com:80`

In centralized configuration mode, the Agent URL is used to populate the Agent Profile for services, such as notifications.

Procedure 4.6. To Install the Web Policy Agent into Apache Virtual Hosts

This procedure assumes you have installed a web policy agent into the default root configuration of your Apache HTTP Server installation, with configuration in `/web_agents/apache24_agent/instances/agent_1`. To install into a virtual host, copy this configuration folder, modify required settings, and enable the web policy agent in the virtual host configuration file.

1. Shut down the Apache server where you plan to install the agent.

2. Locate the web policy agent configuration instance to duplicate, and make a copy, for example `agent_2`:

   - UNIX example:

   ```
   $ cd /web_agents/apache24_agent/instances
   $ cp -r agent_1 agent_2
   ```

   - Windows example:

   ```
   c:\> cd c:\web_agents\apache24_agent\instances
   c:\path\to\web_agents\apache24_agent\instances> xcopy /E /I agent_1 agent_2
   ```

3. Give the user that runs the virtual host modify privileges to the new instance folder. The following examples demonstrate giving privileges to the `agent_2` configuration instance to a user named *apache*:

   - UNIX example:

   ```
   $ cd /web_agents/apache24_agent/instances
   $ chown -hR apache agent_2
   ```

   - Windows example:

   ```
   c:\> cd c:\web_agents\apache24_agent\instances
   c:\path\to\web_agents\apache24_agent\instances> icacls "agent_2" /grant apache:M
   ```

4. In the new instance folder, edit the `/config/agent.conf` configuration file as follows:

a. Alter the value of `com.sun.identity.agents.config.username` to be the name of the agent profile you created in OpenAM for the virtual host.

b. If you used a different password when creating the new agent profile in OpenAM, you will need to configure the encryption key and password value in the agent configuration file.

   Generate a new signing key, by running **agentadmin --k**.

   Use the generated encryption key to encrypt the new password, by running `agentadmin --p`, specifying the encryption key and the new password:

   - UNIX example:

     ```
     $ ./agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" "newpassword"
     Encrypted password value: 07bJOSeM/G8ydO4=
     ```

   - Windows example:

     ```
     C:\path\to\web_agents\apache24_agent\bin>
       agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" "newpassword"
     Encrypted password value: 07bJOSeM/G8ydO4=
     ```

   In the agent configuration file of the new instance, set the following properties:

   - `com.sun.identity.agents.config.key` to be the generated encryption key value.

     For example:

     com.sun.identity.agents.config.key = YWM0OThlMTQtMzMxOS05Nw==

   - `com.sun.identity.agents.config.password` to be the generated encrypted password value.

     For example:

     com.sun.identity.agents.config.password = 07bJOSeM/G8ydO4=

c. Replace any references to the original instance directory with the new instance directory. For example, replace the string `agent_1` with `agent_2` wherever it occurs in the configuration file.

   Configuration options that are likely to require alterations include:

   - `com.sun.identity.agents.config.local.logfile`

- `com.sun.identity.agents.config.local.audit.logfile`

d. Replace any references to the original website being protected with the new website being protected. For example, replace `http://www.example.com:80/amagent` with `http://customers.example.com:80/amagent`.

   Configuration options that are likely to require alterations include:

   - `com.sun.identity.client.notification.url`
   - `com.sun.identity.agents.config.agenturi.prefix`
   - `com.sun.identity.agents.config.fqdn.default`

e. Save and close the configuration file.

5. Edit the Apache HTTP Server configuration file. This is the same file specified when installing the web policy agent into the default Apache website. For example, `/etc/httpd/conf/httpd.conf`.

   a. At the end of the file the installer will have added three new lines of settings, for example:

```
LoadModule amagent_module /web_agents/apache24_agent/lib/mod_openam.so
AmAgent On
AmAgentConf /web_agents/apache24_agent/bin/../instances/agent_1/config/agent.conf
```

   Leave the first line, `LoadModule ...`, and move the other two lines into the virtual host configuration element of the default site, for example:

```
<VirtualHost *:80>
# This first-listed virtual host is also the default for *:80
ServerName www.example.com
ServerAlias example.com
DocumentRoot "/var/www/html"
AmAgent On
AmAgentConf /web_agents/apache24_agent/instances/agent_1/config/agent.conf
</VirtualHost>
```

   b. Copy the same two lines into the new virtual host, and replace `agent_1` with the new agent configuration instance folder, for example `agent_2`:

```
<VirtualHost *:80>
ServerName customers.example.com
DocumentRoot "/var/www/customers"
AmAgent On
AmAgentConf /web_agents/apache24_agent/instances/agent_2/config/agent.conf
</VirtualHost>
```

Tip

If the new virtual host configuration is in a separate file, copy the two configuration lines into the `VirtualHost` element within that file.

6. Save and close the Apache HTTP Server configuration file.
7. Start the Apache HTTP server in which you installed the web policy agent.

Procedure 4.7. To Check the Policy Agent Installation

1. Check the Apache HTTP server error log after you start the server to make sure startup completed successfully:

```
[Tue Sep 08 15:51:27.667625 2015] AH00163:
 Apache/2.4.6 (CentOS) OpenAM Web Agent/4 configured
 -- resuming normal operations
```

2. Check the `/web_agents/apache24_agent/instances/Agent_1/logs/debug/debug.log` file to verify that no errors occurred on startup. Expected output should resemble the following:

```
2015-09-08 16:02:24.573 -0700 INFO [0x7f7470064840:5748]

######################################################
  OpenAM Web Agent
  Version: 4
  Revision: 15441
  Build date: Aug 29 2015 02:48:01
######################################################
```

3. If you have a policy configured, you can test your policy agent. For example, try to browse to a resource that your policy agent protects. You should be redirected to OpenAM to authenticate, for example, as user `demo`, password *`changeit`*. After you authenticate, OpenAM redirects you back to the resource you tried to access.

4.4 Installing Apache Web Policy Agents Silently

You can run a silent, non-interactive installation by running **agentadmin --s**, along with arguments used to configure the instance.

Tip

Check that you have completed the prerequisite steps before proceeding. See Section 4.1, "Before You Install".

The required arguments, and the order in which to specify them are:

Web server configuration file
: Enter the full path to the Apache HTTP server configuration file. The installer modifies this file to include the web policy agent configuration and module.

OpenAM URL
: Enter the full URL of the OpenAM instance the web policy agents will be using. Ensure the deployment URI is specified.

Agent URL
: Enter the full URL of the server the agent is running on.

Realm
: Enter the OpenAM realm containing the agent profile.

Agent profile name
: Enter the name given to the agent profile created in OpenAM.

Agent profile password
: Enter the full path to the file containing the agent profile password.

`--changeOwner`
: To have the installer change the ownership of created directories to be the same User and Group as specified in the Apache configuration, specify the optional `--changeOwner` switch.

`--acceptLicence`
: You can suppress the license agreement prompt during a silent, non-interactive install by including the `--acceptLicence` parameter. The inclusion of the option indicates that you have read and accepted the terms stated in the license. To view the license agreement, open `/path/to/web_agents/`*`agent_type`*`/legal/Forgerock_License.txt`.

`--forceInstall`
: Optionally have the installer proceed with a silent installation even if it cannot connect to the specified OpenAM server during installation, rather than exiting.

For example:

```
$ agentadmin --s \
  "/etc/httpd/conf/httpd.conf" \
  "http://openam.example.com:8080/openam" \
  "http://www.example.com:80" \
  "/" \
  "webagent4" \
  "/tmp/pwd.txt" \
  --changeOwner \
  --acceptLicence

OpenAM Web Agent for Apache Server installation.

Validating...
Validating... Success.
Cleaning up validation data...
Creating configuration...
Installation complete.
```

4.5 Removing Apache Web Policy Agents

Procedure 4.8. To remove Web Policy Agents from Apache HTTP Server

1. Shut down the Apache server where the agent is installed.

2. Run **agentadmin --l** to output a list of the installed web policy agent configuration instances.

   Make a note of the ID value of the configuration instance you want to remove.

3. Run **agentadmin --r**, and specify the ID of the web policy agent configuration instance to remove. A warning is displayed. Type `yes` to proceed with removing the configuration instance.

   ```
   $ ./agentadmin --r agent_3

   Warning! This procedure will remove all OpenAM Web Agent references from
   a Web server configuration. In case you are running OpenAM Web Agent in a
   multi-virtualhost mode, an uninstallation must be carried out manually.

   Continue (yes/no): [no]: yes

   Removing agent_3 configuration...
   Removing agent_3 configuration... Done.
   ```

4. Restart the Apache HTTP Server.

Chapter 5

Installing Web Policy Agents in Microsoft IIS

This chapter covers prerequisites and installation procedures for Web Policy Agents 4 into Microsoft Internet Information Services (*IIS*) 7 and 8.

5.1 Before You Install

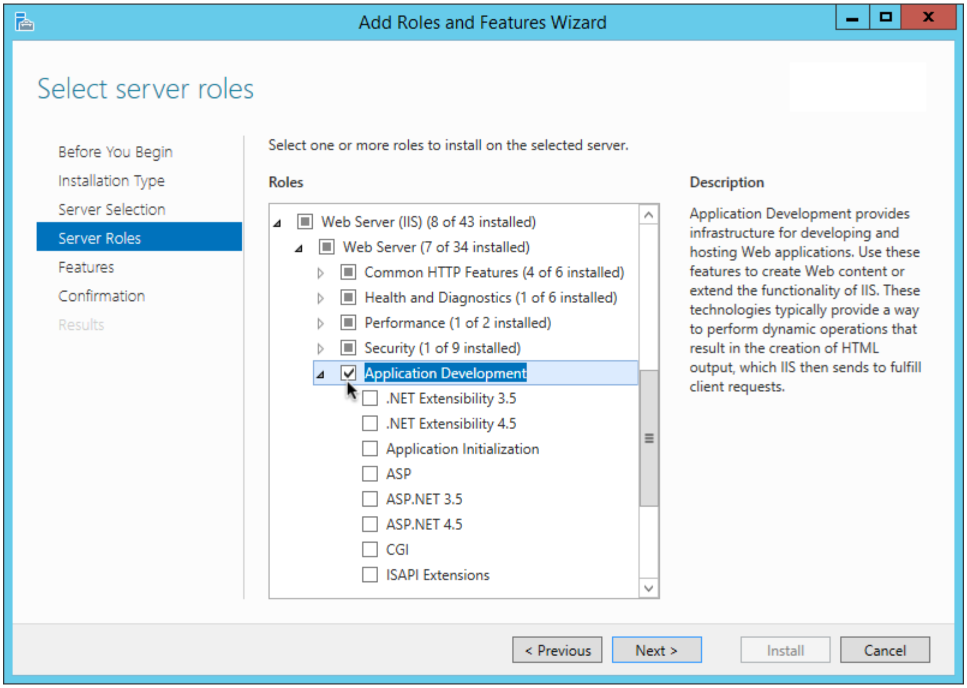
This section describes the prerequisite steps you should take before installing the web policy agents into IIS servers.

- Ensure OpenAM is installed and running, so that you can contact OpenAM from the system running the policy agent.
- Create a profile for your policy agent as described in *Configuring Web Policy Agent Profiles*.
- Create at least one policy in OpenAM to protect resources with the agent, as described in the section on *Configuring Policies*. Consider creating a simple policy, such as a policy that allows only authenticated users to access your resources. This allows you to test your policy agent after installation.
- If the OpenAM server you will be connecting to uses SSL, you must install OpenSSL on the agent machine.

Ensure the OpenSSL libraries `libeay32.dll` and `ssleay32.dll` are available in the `lib` folder of your agent installation, for example `c:\path\to\web_agents\iis_agent\lib\`.

- Web policy agents require that the *Application Development* component is installed alongside the core IIS services. Application Development is an optional component of the IIS web server. The component provides required infrastructure for hosting web applications.

Figure 5.1. Adding the Application Development Component to IIS



Select server roles

Select one or more roles to install on the selected server.

Roles

- ■ Web Server (IIS) (8 of 43 installed)
  - ■ Web Server (7 of 34 installed)
    - ■ Common HTTP Features (4 of 6 installed)
    - ■ Health and Diagnostics (1 of 6 installed)
    - ■ Performance (1 of 2 installed)
    - ■ Security (1 of 9 installed)
    - ☑ Application Development
      - □ .NET Extensibility 3.5
      - □ .NET Extensibility 4.5
      - □ Application Initialization
      - □ ASP
      - □ ASP.NET 3.5
      - □ ASP.NET 4.5
      - □ CGI
      - □ ISAPI Extensions

Description

Application Development provides infrastructure for developing and hosting Web applications. Use these features to create Web content or extend the functionality of IIS. These technologies typically provide a way to perform dynamic operations that result in the creation of HTML output, which IIS then sends to fulfill client requests.

- See the OpenAM *Installation Guide* section, *Obtaining OpenAM Software* to determine which version of the agent to download, and download the agent. Also, verify the checksum of the file you download against the checksum posted on the download page.

  Unzip the file in the directory where you plan to install the web policy agent. The agent you install stores its configuration and logs under this directory.

  When you unzip the policy agent `.zip` download, you find the following directories:

`bin`
: The installation and configuration program **agentadmin**.

`config`
: Configuration templates used by the **agentadmin** command during installation.

`instances`
: Configuration files, and audit and debug logs for individual instances of the web policy agents will be created here. The folder is empty when first extracted.

`legal`
: Contains licensing information including third-party licenses.

`lib`
: Shared libraries used by the policy agent.

`log`
: Location for log files written during installation. The folder is empty when first extracted.

5.2 Installing IIS Web Policy Agents

Complete the following procedures to install Web Policy Agents 4 into Apache HTTP Servers.

> **Tip**
>
> Check that you have completed the prerequisite steps before proceeding. See Section 5.1, "Before You Install".

Procedure 5.1. To Create the Agent Profile

Regardless of whether you store configurations centrally in OpenAM or locally with your agents, the agent requires a profile so that it can connect to and communicate with OpenAM.

1. In the OpenAM console, browse to Realms > *`Realm Name`* > Agents > Web, and then click the New... button in the Agent table.

2. Complete the web form using the following hints.

Name
: The name for the agent profile used when you install the agent

Password
: Password the agent uses to authenticate to OpenAM

Configuration
: Centralized configurations are stored in the OpenAM configuration store. You can manage the centralized configuration through the OpenAM console. Local configurations are stored in a file alongside the agent.

Server URL
: The full URL to an OpenAM instance, or if OpenAM is deployed in a site configuration (behind a load balancer) then the site URL

  In centralized configuration mode, the Server URL is used to populate the agent profile for services, such as Login, Logout, Naming, and Cross Domain SSO.

Agent URL
: The URL to the web agent application, such as `http://www.example.com:80`

  In centralized configuration mode, the Agent URL is used to populate the Agent Profile for services, such as notifications.

Procedure 5.2. To Create a Password File

1. Create a text file containing only the password specified when creating the agent profile.

   UNIX example:

   ```
   $ echo password > /tmp/pwd.txt
   ```

   Windows example:

   ```
   C:\> echo password > pwd.txt
   ```

2. Protect the password file you create as appropriate for your operating system:

   UNIX example:

   ```
   $ chmod 400 /tmp/pwd.txt
   ```

   Windows example:

In Windows Explorer, right-click the created password file, for example `pwd.txt`, select Read-Only, and then click OK.

Procedure 5.3. To Install the Policy Agent into IIS

1. Log on to Windows as a user with administrator privileges.

2. Make sure OpenAM is running.

3. Run **agentadmin.exe** with the `--i` switch to install the agent. You will be prompted to read and accept the software license agreement for the agent installation.

   ```
   c:\> cd web_agents\iis_agent\bin
   c:\web_agents\iis_agent\bin> agentadmin.exe --i
   ```

4. When prompted for information, enter the inputs appropriate for your deployment.

   Tip

   You can cancel web policy agent installation at anytime by pressing **CTRL+C**

   a. Enter the ID number of the IIS site in which to install the web policy agent.

   ```
   IIS Server Site configuration:

   Number of Sites: 2
   id: 1   name: "DEFAULT WEB SITE"
   id: 2   name: "CUSTOMERPORTAL"

   Enter IIS Server Site identification number.
   [ q or 'ctrl+c' to exit ]
   Site id: 2
   ```

   b. The installer can import settings from an existing web policy agent into the new installation and skips prompts for any values present in the existing configuration file. You will be required to re-enter the agent profile password.

   Enter the full path to an existing agent configuration file to import the settings, or press **Enter** to skip the import.

```
To set properties from an existing configuration enter path to file
[ q or 'ctrl+c' to exit, return to ignore ]
Existing agent.conf file:
```

c. Enter the full URL of the OpenAM instance the web policy agents will be using. Ensure the deployment URI is specified.

```
Enter the URL where the OpenAM server is running. Please include the
deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ q or 'ctrl+c' to exit ]
OpenAM server URL: http://openam.example.com:8080/openam
```

d. Enter the full URL of the site the agent will be running in.

```
Enter the Agent URL as shown below:
(http://agent.sample.com:1234)
[ q or 'ctrl+c' to exit ]
Agent URL: http://customers.example.com:80
```

e. Enter the name given to the agent profile created in OpenAM.

```
Enter the Agent profile name
[ q or 'ctrl+c' to exit ]
Agent Profile name: iisagent
```

f. Enter the OpenAM realm containing the agent profile.

```
Enter the Agent realm/organization
[ q or 'ctrl+c' to exit ]
Agent realm/organization name: [/]: /
```

g. Enter the full path to the file containing the agent profile password created earlier.

```
Enter the path to a file that contains the password to be used
for identifying the Agent
[ q or 'ctrl+c' to exit ]
The path to the password file: c:\pwd.txt
```

h. The installer displays a summary of the configuration settings you specified.

- If a setting is incorrect, type `no`, or press **Enter**. The installer loops through the configuration prompts using your provided settings as the default. Press **Enter** to accept each one, or enter a replacement setting.

- If the settings are correct, type `yes` to proceed with installation.

```
Installation parameters:

   OpenAM URL: http://openam.example.com:8080/openam
   Agent URL: http://customers.example.com:80
   Agent Profile name: iisagent
   Agent realm/organization name: /
   Agent Profile password source: c:\pwd.txt

Confirm configuration (yes/no): [no]: yes
Validating...
Validating... Success.
Cleaning up validation data...
Creating configuration...
Installation complete.
```

Upon successful completion, the installer adds the agent as a module to the IIS site configuration.

The installer also sets up configuration and log directories for the agent instance. Each agent instance that you install on the system has its own numbered configuration and logs directory. The first agent's configuration and logs are located under the directory `web_agents\iis_agent\instances\agent_1\`.

Note

The installer grants full access permissions on the created instance folder to the user that the selected IIS site is running under, so that log files can be written correctly.

The configuration files and log locations are as follows:

`config/agent.conf`
Contains the bootstrap properties the web policy agent requires to connect to OpenAM and download its configuration. Also contains properties that are only used if you configure the web policy agent to use local configuration.

`logs/audit/`
Operational audit log directory, only used if remote logging to OpenAM is disabled.

`logs/debug/`
Debug directory where the `amAgent` debug file resides. Useful in troubleshooting policy agent issues.

5.3 Installing IIS Web Policy Agents Silently

You can run a silent, non-interactive installation by running **agentadmin.exe --s**, along with arguments used to configure the instance.

Tip

Check that you have completed the prerequisite steps before proceeding. See Section 5.1, "Before You Install".

The required arguments, and the order in which to specify them are:

Web server configuration file
: Enter the ID number of the IIS site in which to install the web policy agent.

Tip

To list the sites in an IIS server, run **agentadmin.exe --n**:

OpenAM URL
: Enter the full URL of the OpenAM instance the web policy agents will be using. Ensure the deployment URI is specified.

Agent URL
: Enter the full URL of the IIS site the agent will be running on.

Realm
: Enter the OpenAM realm containing the agent profile.

Agent profile name
: Enter the name given to the agent profile created in OpenAM.

Agent profile password
: Enter the full path to the file containing the agent profile password.

`--changeOwner`
: Optionally have the installer change the ownership of created directories to be the same user that is running the selected IIS site.

`--acceptLicence`
: You can suppress the license agreement prompt during a silent, non-interactive install by including the `--acceptLicence` parameter. The

inclusion of the option indicates that you have read and accepted the terms stated in the license. To view the license agreement, open `/path/to/web_agents/`*`agent_type`*`/legal/Forgerock_License.txt`.

`--forceInstall`

Add this optional switch to have the installer proceed with a silent installation even if it cannot connect to the specified OpenAM server during installation, rather than exiting.

For example:

```
c:\web_agents\iis_agent\bin> agentadmin.exe --s ^
  "1" ^
  "http://openam.example.com:8080/openam" ^
  "http://iis.example.com:80" ^
  "/" ^
  "iisagent" ^
  "c:\pwd.txt" ^
  --changeOwner ^
  --acceptLicence

OpenAM Web Agent for IIS Server installation.

Validating...
Validating... Success.
Cleaning up validation data...
Creating configuration...
Installation complete.
```

5.4 Managing IIS Web Policy Agents

This section explains how to disable, enable, and remove web policy agents that are in an IIS site, and how to completely uninstall web policy agents from IIS.

Procedure 5.4. To disable and enable a web policy agent in an IIS site

1. Log on to Windows as a user with administrator privileges.

2. Run **agentadmin.exe --l** to output a list of the installed web policy agent configuration instances.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --l
   OpenAM Web Agent configuration instances:

      id:            agent_1
      configuration: c:\web_agents\iis_agent\bin\..\instances\agent_1
      server/site:   2
   ```

   Make a note of the ID value of the configuration instance you want to disable or enable.

3. Perform one of the following steps:

   - To disable the web policy agent in a site, run **agentadmin.exe --d**, and specify the ID of the web policy agent configuration instance to disable.

     ```
     c:\web_agents\iis_agent\bin> agentadmin.exe --d agent_1

     Disabling agent_1 configuration...
     Disabling agent_1 configuration... Done.
     ```

   - To enable the web policy agent in a site, run **agentadmin.exe --e**, and specify the ID of the web policy agent configuration instance to enable.

     ```
     c:\web_agents\iis_agent\bin> agentadmin.exe --e agent_1

     Enabling agent_1 configuration...
     Enabling agent_1 configuration... Done.
     ```

Procedure 5.5. To remove a web policy agent from an IIS site

1. Log on to Windows as a user with administrator privileges.

2. Run **agentadmin.exe --l** to output a list of the installed web policy agent configuration instances.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --l
   OpenAM Web Agent configuration instances:

      id:            agent_1
      configuration: c:\web_agents\iis_agent\bin\..\instances\agent_1
      server/site:   2
   ```

   Make a note of the ID value of the configuration instance you want to remove.

3. Run **agentadmin.exe --r**, and specify the ID of the web policy agent configuration instance to remove.

   ```
   c:\web_agents\iis_agent\bin> agentadmin.exe --r agent_1

   Removing agent_1 configuration...
   Removing agent_1 configuration... Done.
   ```

Procedure 5.6. To remove web policy agents from IIS

1. Log on to Windows as a user with administrator privileges.

2. Run **agentadmin --g**. A warning is displayed. Type yes to proceed with removing the configuration instance.

```
c:\web_agents\iis_agent\bin> agentadmin.exe --g

Warning! This procedure will remove all OpenAM Web Agent references from
IIS Server configuration.

Continue (yes/no): [no]: yes

Removing agent module from IIS Server configuration...
Removing agent module from IIS Server configuration... Done.
```

5.5 Enable IIS Basic Authentication and Password Replay Support

The IIS web policy agent now supports IIS basic authentication and password replay. You must use the appropriate software versions.

Given the proper configuration and with Active Directory as a user data store for OpenAM, the IIS web policy agent can provide access to the IIS server variables. The instructions for configuring the capability follow in this section, though you should read the section in full, also paying attention to the required workarounds for Microsoft issues.

When configured as described, the policy agent requests IIS server variable values from OpenAM, which gets them from Active Directory. The policy agent then sets the values in HTTP headers so that they can be accessed by your application.

The following IIS server variables all take the same value when set: `REMOTE_USER`, `AUTH_USER`, and `LOGON_USER`. The policy agent either sets all three, or does not set any of them.

When you enable Logon and Impersonation in the console (`com.sun.identity.agents.config.iis.logonuser=true` in the policy agent configuration), the policy agent performs Windows logon and sets the user impersonation token in the IIS session context.

When you enable Show Password in HTTP Header in the console (`com.sun.identity.agents.config.iis.password.header=true` in the policy agent configuration), the policy agent adds it in the `USER_PASSWORD` header.

The policy agent does not modify any other IIS server variables related to the authenticated user's session.

The policy agent works best with IIS running in Integrated, not Classic mode. In Classic mode, you cannot share sessions between the policy agent and another .NET application, so Logon and Impersonation are not operative. Furthermore IIS in Classic mode treats all modules as ISAPI extensions, and request processing is affected. It is therefore strongly recommended that you run IIS in Integrated mode:

- For Microsoft Office integration, you must use Microsoft Office 2007 SP2 or later.
- For Microsoft SharePoint integration, you must use Microsoft SharePoint Server 2007 SP2 or later.

You must also apply workarounds as described for the following Microsoft issues.

Microsoft Support Issue: 841215
Link: http://support.microsoft.com/kb/841215

Description: Error message when you try to connect to a Windows SharePoint document library: "System error 5 has occurred".

Summary: Enable Basic Authentication on the client computer.

Microsoft Support Issue: 870853
Link: http://support.microsoft.com/kb/870853

Description: Office 2003 and 2007 Office documents open read-only in Internet Explorer.

Summary: Add registry keys as described in Microsoft's support document.

Microsoft Support Issue: 928692
Link: http://support.microsoft.com/kb/928692

Description: Error message when you open a Web site by using Basic authentication in Expression Web on a computer that is running Windows Vista: "The folder name is not valid".

Summary: Edit the registry as described in Microsoft's support document.

Microsoft Support Issue: 932118
Link: http://support.microsoft.com/kb/932118

Description: Persistent cookies are not shared between Internet Explorer and Office applications.

Summary: Add the web site the list of trusted sites.

Microsoft Support Issue: 943280
Link: http://support.microsoft.com/kb/943280

Description: Prompt for Credentials When Accessing FQDN Sites From a Windows Vista or Windows 7 Computer.

Summary: Edit the registry as described in Microsoft's support document.

Microsoft Support Issue: 968851
Link: http://support.microsoft.com/kb/968851

Description: SharePoint Server 2007 Cumulative Update Server Hotfix Package (MOSS server-package): April 30, 2009.

Summary: Apply the fix from Microsoft if you use SharePoint.

Microsoft Support Issue: 2123563
Link: http://support.microsoft.com/kb/2123563

Description: You cannot open Office file types directly from a server that supports only Basic authentication over a non-SSL connection.

Summary: Enable SSL encryption on the web server.

Procedure 5.7. To Configure IIS Basic Authentication and Password Replay Support

Follow these steps:

1. Generate and store an encryption key:

   a. Generate the key using `com.sun.identity.common.DESGenKey` using the .jars where you deployed OpenAM, as in the following example. The Java command below is broken out into multiple lines for display purposes only:

   ```
   $ cd /tomcat/webapps/openam/WEB-INF/lib
   $ java -cp forgerock-util-3.0.0.jar:openam-core-13.jar:\
      openam-shared-13.jar com.sun.identity.common.DESGenKey
   Key ==> sxVoaDRAN0o=
   ```

   Windows users should use semi-colons (";"), instead of colons (":") in the commands. The Java command below is broken out into multiple lines for display purposes only:

   ```
   c:\> cd \tomcat\webapps\openam\WEB-INF\lib
   c:\> java -cp forgerock-util-3.0.0.jar;openam-core-13.jar; ^
       openam-shared-13.jar com.sun.identity.common.DESGenKey
   Key ==> sxVoaDRAN0o=
   ```

   b. In the OpenAM console navigate to Realms > *Realm Name* > Agents > Web > *Agent Name* > Advanced > Microsoft IIS Server > Replay Password Key (property name: `com.sun.identity.agents.config.replaypasswd.key`), enter the generated key, and then click Save.

   c. In the OpenAM console, navigate to Configuration > Servers and Sites > *Server Name* > Advanced > Add..., then add a property `com.sun.am.`

replaypasswd.key with the key you generated as the value, and then click Save.

2. In the OpenAM console, navigate to Realms > *Realm Name* > Authentication > Settings > Post Authentication Processing > Authentication Post Processing Classes, then add the class com.sun.identity.authentication.spi.ReplayPasswd, and then click Save.

3. If you require Windows logon, or you need to use basic authentication with SharePoint or OWA, then you must configure Active Directory as a user data store, and you must configure the IIS policy agent profile User ID Parameter and User ID Parameter Type so that the policy agent requests OpenAM to provide the appropriate account information from Active Directory in its policy response.

   Skip this step if you do not use SharePoint or OWA and no Windows logon is required.

   Make sure OpenAM data store is configured to use Active Directory as the user data store.

   In the OpenAM console under Realms > *Realm Name* > Agents > Web > *Agent Name* > OpenAM Services > Policy Client Service, set User ID Parameter and User ID Parameter Type, and then save your work. For example if the real username for Windows domain logon in Active Directory is stored on the sAMAccountName attribute, then set the User ID Parameter to sAMAccountName, and the User ID Parameter Type to LDAP.

   Setting the User ID Parameter Type to LDAP causes the policy agent to request that OpenAM get the value of the User ID Parameter attribute from the data store, in this case, Active Directory. Given that information, the policy agent can set the HTTP headers REMOTE_USER, AUTH_USER, or LOGON_USER and USER_PASSWORD with Active Directory attribute values suitable for Windows logon, setting the remote user, and so forth.

4. To set the encrypted password in the AUTH_PASSWORD header, browse in the OpenAM console to Realms > *Realm Name* > Agents > Web > *Agent Name* > Advanced > Microsoft IIS Server, then select Show Password in HTTP Header, and then click Save.

5. To have the agent perform Windows logon (for user token impersonation), browse in the OpenAM console to Realms > *Realm Name* > Agents > Web > *Agent Name* > Advanced > Microsoft IIS Server, then select Logon and Impersonation, and then click Save.

6. In the OpenAM console, navigate to Realms > *Realm Name* > Agents > Web > *Agent Name* > Advanced > Microsoft IIS Server, then set Authentication Type to basic, and then click Save.

7. To use the agent with SharePoint or Microsoft Office, configure OpenAM to persist the authentication cookie. For details, see "Hints for the Persistent Cookie Module" in the *OpenAM Administration Guide*.

Chapter 6

Troubleshooting

This chapter offers solutions to issues during installation of OpenAM policy agents.

Solutions to Common Issues

This section offers solutions to common problems when installing OpenAM policy agents:

Q: I am trying to install a policy agent on Windows, which will connect to an OpenAM server running over HTTPS, but the installer reports the following:

```
init_ssl(): ssleay32.dll is not available (error: 87)
init_ssl(): libeay32.dll is not available (error: 87)
```

A: If OpenSSL is correctly installed and you are using a Windows 7 or Windows Server 2008 R2 system, apply the update provided in Microsoft knowledge base article KB2533623. See Microsoft Security Advisory: Insecure library loading could allow remote code execution.

Q: I am trying to install the policy agent on SELinux and I am getting error messages after installation. What happened?

A: SELinux must be properly configured to connect the web policy agent and OpenAM nodes. Either re-configure SELinux or disable it, then reinstall the policy agent.

Q: My Apache HTTP server is not using port 80. But when I install the web policy agent it defaults to port 80. How do I fix this?

A: You probably set `ServerName` in the Apache HTTP Server configuration to the host name, but did not specify the port number.

Instead you must set both the host name and port number for `ServerName` in the configuration. For example, if you have Apache HTTP Server configured to listen on port 8080, then set `ServerName` appropriately as in the following excerpt:

```
<VirtualHost *:8080>
ServerName www.localhost.example:8080
```

Q: My web server and web policy agent are installed as root, and the agent cannot rotate logs. I am seeing this error:

```
Could not rotate log file ... (error: 13)
```

What should I do?

A: First, avoid installing the web server (and therefore also the web policy agent) as root, but instead create a web server user and install as that user.

If however you cannot avoid installing the web server and policy agent as root, the you must give all users read and write permissions to the `logs/` directory under the agent instance directory (`/web_agents/`*`agent_version`*`/instances/agent_`*`nnn`*`/logs/`). Otherwise, the web policy agent fails to rotate log files with the error you observed.

Q: How do I increase security against possible phishing attacks through open redirect?

A: You can specify a list of valid URL resources against which OpenAM validates the `goto` and `gotoOnFail` URL using the Valid `goto` URL Resource service.

OpenAM only redirects a user if the `goto` and `gotoOnFail` URL matches any of the resources specified in this setting. If no setting is present, it is assumed that the `goto` and `gotoOnFail` URL is valid.

To set the Valid `goto` URL Resources, use the OpenAM console, and navigate to Realms > *`Realm Name`* > Services. Click Add, select Validation Service, and then add one or more valid `goto` URLs.

You can use the "*" wildcard to define resources, where "*" matches all characters except "?". For example, you can use the wildcards, such as `https://website.example.com/*` or `https://website.example.com/*?*`. For more specific patterns, use resource names with wildcards as described in the procedure, *Configuring Valid goto URL Resources*.

Command-Line Tool Reference

Table of Contents

agentadmin

agentadmin — manage OpenAM web policy agent installation

agentadmin

`agentadmin` {options}

1 Description

This command manages OpenAM policy agent installations.

2 Options

The following options are supported:

`--i`

Perform an interactive install of a new agent instance.

Usage: **agentadmin --i**

For more information, see:

- Section 4.2, "Installing Apache Web Policy Agents"
- Section 5.2, "Installing IIS Web Policy Agents"

`--s`

Perform a silent, non-interactive install of a new agent instance.

Usage: **agentadmin --s *`web-server-config-file`* *`openam-url`* *`agent-url`* *`realm`* *`agent-profile-name`* *`agent-profile-password`* [--changeOwner] [--acceptLicense] [--forceInstall]**

`web-server-config-file`

When installing in Apache HTTP Server, enter the full path to the Apache HTTP server configuration file. The installer modifies this file to include the web policy agent configuration and module.

When installing in Microsoft IIS, enter the ID number of the IIS site in which to install the web policy agent. To list the available sites in an IIS server and the relevant ID numbers, run **agentadmin.exe --n**.

`openam-url`

Enter the full URL of the OpenAM instance that the web policy agents will use. Ensure the deployment URI is specified.

Example:

```
https://openam.example.com:8443/openam
```

agent-url
: Enter the full URL of the server on which the agent is running.

Example:

```
http://www.example.com:80
```

realm
: Enter the OpenAM realm containing the agent profile.

agent-profile-name
: Enter the name of the agent profile in OpenAM.

agent-profile-password
: Enter the full path to the agent profile password file.

`--changeOwner`
: Use this option to change the ownership of the created directories to be the same user and group as specified in the Apache HTTP Server configuration, or the user that is running the selected IIS site.

`--acceptLicense`
: When you run certain commands, you will be prompted to read and accept the software license agreement. You can suppress the license agreement prompt by including the optional `--acceptLicence` parameter. Specifying this options indicates that you have read and accepted the terms stated in the license.

To view the license agreement, open `/path/to/web_agents/agent_type/legal/Forgerock_License.txt`.

`--forceInstall`
: Add this option to proceed with a silent installation even if it cannot connect to the specified OpenAM server during installation, rather than exiting.

For more information, see:

- Section 4.4, "Installing Apache Web Policy Agents Silently"
- Section 5.3, "Installing IIS Web Policy Agents Silently"

`--n`
: List the sites available in an IIS server.

Example:

```
c:\web_agents\iis_agent\bin> agentadmin.exe --n

 IIS Server Site configuration:

 Number of Sites: 2

 id: 1   name: "DEFAULT WEB SITE"
 id: 2   name: "CUSTOMERPORTAL"
```

`--l`

List existing configured agent instances.

Usage: **agentadmin --l**

Example:

```
$ ./agentadmin --l
OpenAM Web Agent configuration instances:

 id:            agent_1
 configuration: /opt/web_agents/apache24_agent/bin/../instances/agent_1
 server/site:   /etc/httpd/conf/httpd.conf

 id:            agent_2
 configuration: /opt/web_agents/apache24_agent/bin/../instances/agent_2
 server/site:   /etc/httpd/conf/httpd.conf

 id:            agent_3
 configuration: /opt/web_agents/apache24_agent/bin/../instances/agent_3
 server/site:   /etc/httpd/conf/httpd.conf
```

`--r`

Remove an existing agent instance.

Usage: **agentadmin --r *`agent-instance`***

`agent-instance`

The ID of the web policy agent configuration instance to remove.

Respond `yes` when prompted to confirm removal.

For more information, see:

- Section 4.5, "Removing Apache Web Policy Agents"
- Section 5.4, "Managing IIS Web Policy Agents"

`--k`

Generate a new signing key.

Usage: **agentadmin --k**

Examples:

- UNIX:

```
$ cd /web_agents/apache24_agent/bin/
$ ./agentadmin --k
Encryption key value: YWM0OThlMTQtMzMxOS05Nw==
```

- Windows:

```
C:\> cd web_agents\apache24_agent\bin
C:\web_agents\apache24_agent\bin> agentadmin --k
Encryption key value: YWM0OThlMTQtMzMxOS05Nw==
```

For more information, see Encryption Properties.

`--p`

Use a generated encryption key to encrypt a new password.

Usage: **agentadmin --p** ***`encryption-key password`***

`encryption-key`

An encryption key, generated by the **agentadmin --k** command.

`password`

The password to encrypt.

Examples:

- UNIX:

```
$ ./agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" "newpassword"
Encrypted password value: 07bJOSeM/G8ydO4=
```

- Windows:

```
C:\web_agents\apache24_agent\bin> agentadmin --p "YWM0OThlMTQtMzMxOS05Nw==" "newpassword"
Encrypted password value: 07bJOSeM/G8ydO4=
```

For more information, see Encryption Properties.

`--v`

Display **agentadmin** build and version information.

Index