Technical Note: Precautions Against Cookie Hijacking in an Access Manager Deployment

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–7849–10
February 14, 2006

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun$^{\text{TM}}$ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

070214@16599

Contents

Precautions Against Session-Cookie Hijacking in an Access Manager Deployment

This technical note provides information about precautions you can take against specific security threats related to session-cookie hijacking in an Access Manager deployment. A common concern for administrators who want to restrict access to web-based applications in an Access Manager deployment is that hackers might use applications, referred to as "rogue" or "untrusted," to hijack session cookies. This document describes the threat posed by this hacking technique and provides configuration steps you can perform to guard against this threat, as described in the following sections:

- "Defining Key Cookie Hijacking Security Issues" on page 5
- "Access Manager Solution: Cookie Hijacking Security Issues" on page 9
- "Implementing the Access Manager Solution for Cookie Hijacking Security Issues" on page 12

The tasks presented in this document apply to a deployment with the following components:

- Starting with Access Manager 7.0
- Starting with Policy Agent 2.2

  The Policy Agent 2.2 software set includes two types of agents: J2EE agents and web agents. Configuring these two agent types requires slightly different instructions, which are provided where appropriate.

Defining Key Cookie Hijacking Security Issues

The tasks presented in this document address specific security issues related to session-cookie hijacking. This section defines those security issues.

The term "cookie hijacking" simply refers to a situation where an impostor (a hacker, perhaps using an untrusted application) gains unauthorized access to cookies. Therefore, cookie hijacking, by itself, does not refer to unauthorized access to protected web resources. When the cookies being hijacked are session cookies, then cookie hijacking potentially increases the threat of unauthorized access to protected web resources, depending upon how the system is configured.

This section provides background information about specific security issues related to session-cookie hijacking, illustrating how this type of cookie hijacking is possible and how it can potentially lead to unauthorized access of protected web resources. This section provides the underlying basis for understanding how the tasks presented in this document enable Access Manager to handle the specified security issues.

Access Manager provides Single Sign-On (SSO) and Cross Domain Single Sign-On (CDSSO) features, enabling users to seamlessly authenticate across multiple web-based applications. Access Manager is able to provide this seamless authentication by using hypertext transfer protocol (HTTP) or secure hypertext transfer protocol (HTTPS) to set session cookies on users' browsers.

The way Access Manager is configured influences the way it sets the session cookies. Prior to the implementation of the tasks outlined in this document, Access Manager sets session cookies for the entire domain. Therefore, all applications hosted on the domain share the same session cookies. This scenario could enable an untrusted application to intervene and gain access to the session cookies. Specific configuration steps presented in this document address this issue. After you perform the configuration, Access Manager prevents different applications from sharing the same session cookies.

The following list provides some details about possible security issues related to session-cookie hijacking (labels, such as “Security Issue: Insecure Protocol,” are used to make the issues easy to identify and discuss):

Security Issue: Insecure Protocol

An application does not use a secure protocol, such as HTTPS, which makes the session cookie prone to network eavesdropping.

The following security issues could apply if you do not perform the tasks presented in this document.

Security Issue: Shared Session Cookies

All applications share the same HTTP or HTTPS session cookie. This shared session-cookie scenario enables hackers to intervene by using an untrusted application to hijack the session cookie. With the hijacked session cookie, the untrusted application can impersonate the user and access protected web resources.

Security Issue: A Less Secure Application

If a single “less secure” application is hacked, the security of the entire infrastructure is compromised.

Security Issue: Access to User Profile Attributes	The untrusted application can use the session cookie to obtain and possibly modify the profile attributes of the user. If the user has administrative privileges, the application could do much more damage.

Figure 1 illustrates a typical Access Manager deployment within an enterprise. While the figure helps to define security issues related to cookie hijacking, it also helps to define the solution. Therefore, the figure applies to a deployment where the tasks presented subsequently in this document have already been performed.

The deployment illustrated in the figure uses SSO. Moreover, as part of the solution, the Access Manager implementation of CDSSO is enabled. To guard against potential threats posed by cookie hijacking, CDSSO enablement is required regardless of the number of domains involved. Having CDSSO enabled when the deployment involves a single domain might seem counterintuitive, but ultimately security is increased by taking advantage of the CDSSO framework. For other information about the use of CDSSO in this type of deployment, see “Enabling Access Manager to Use Unique SSO Tokens” on page 11.

The numbers in the figure correspond to the numbered explanations that follow. The explanations combine to provide an outline of the process that takes place when a request is made for a protected resource, emphasizing how the SSO token flows between components. The deployment includes a single virtual AuthN (Authentication) and AuthZ (Policy) server (denoted as Access Manager or Identity Provider in the figure), and a number of applications (Service Providers), denoted as Trusted Application and Untrusted Application in the diagram. The Service Providers are usually front-ended with an agent (specifically, an agent from the Policy Agent 2.2 software set) that handles the SSO (AuthN) and Policy (AuthZ).

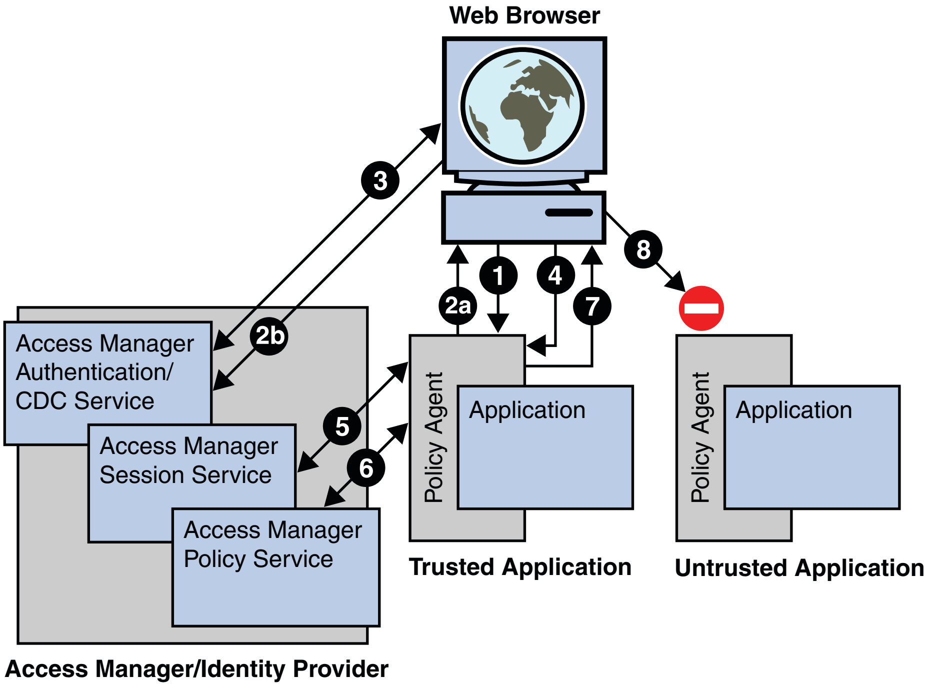
Web Browser
3
8
1
4
2a
7
2b
Access Manager Authentication/ CDC Service
Access Manager Session Service
Access Manager Policy Service
5
6
Policy Agent
Application
Trusted Application
Policy Agent
Application
Untrusted Application
Access Manager/Identity Provider

FIGURE 1 The Role of the Session Cookie in Allowing Access to Protected Web Resources

Note – Figure 1 does not include every detail involved in the flow of the SSO token. The figure provides a simplified view that corresponds to the scope of this document.

1. The user accesses an application through a web browser. The agent, which is an agent from the Policy Agent 2.2 software set, intercepts the request and checks for the user's privilege to access the application. The check is specifically a check for a valid Access Manager SSO token, which is set as a session cookie in the user's browser.
2. Assuming the user does not have a valid SSO token, the agent redirects the browser to Access Manager Authentication Service. The agent also provides its identity using the Liberty AuthN request specification. Since this single redirection is a two step process, it is illustrated in the figure as two substeps: 2A and 2B.
3. Access Manager Authentication Service authenticates the user and, after successful authentication, redirects the user back to the target application with the SSO token as part of the URL query parameter (in a format specified by Liberty AuthN response specification).
4. The agent receives a successful AuthN response from Access Manager Authentication Service. It gets the SSO token and sets it as a session cookie for the host (rather than for the domain) to the browser's request.

5. The agent validates the SSO token with Access Manager Session Service.
6. After a successful SSO token validation in Step 5, the agent then checks the permissions for the user to access the application with Access Manager Policy Service.
7. If permission is granted in Step 6, the user is allowed access to the application.
8. As indicated in the figure with an incomplete connection to Untrusted Application, the same SSO token cannot be used to gain access to an untrusted application. Access Manager denies such access since the SSO token is unique to the application and may not be shared or reissued to other agents or applications.

Access Manager Solution: Cookie Hijacking Security Issues

This section explains how performing the tasks described in this document enables Access Manager to handle the security issues discussed in the preceding section, "Defining Key Cookie Hijacking Security Issues" on page 5.

Note – When applications use a secure protocol such as HTTPS, the SSO Token is not visible to network snooping. This security issue is labeled "Security Issue: Insecure Protocol" in this document. Ensuring that all protected resources use a secure protocol is not a security measure administered using Access Manager, but this a very prudent security measure that you should consider implementing if it is not currently in place.

Access Manager Solution: Shared Session Cookies

The security issue labeled "Security Issue: Shared Session Cookies" in this document pertains to applications sharing the same HTTP or HTTPS session cookie. Access Manager addresses this security threat by issuing a unique SSO token to each Application/Agent after the user has been authenticated. The unique SSO token is referred to as a "restricted token."

The term "Application/Agent," indicates that the restricted token is inextricably connected to the application and to the agent (which specifically refers to an agent from the Policy Agent 2.2 software set). Since each user's SSO token is unique for each Application/Agent, the increased security provided by this scenario prevents an untrusted application, impersonating the user, from accessing other applications. More specifically, since the SSO token (restricted token) assigned to a user (as a part of the user's session) is associated with the agent that did the initial redirection for authentication, all subsequent requests are checked to verify that they are coming from the same agent. Thus, if a hacker tries to use the same restricted token to access another application, a security violation is thrown.

What makes the restricted token "restricted" is not related to the syntax of the token. The syntax of a restricted token is the same as that of a regular SSO token. Instead, a specific constraint is associated with the restricted token. This constraint is what ensures that the restricted token is only used for an application that a given agent protects.

Access Manager Solution: A Less Secure Application

The security issue labeled "Security Issue: A Less Secure Application" in this document pertains to the potential threat of applications that are "less secure." With the Access Manager solution, if one application is somehow compromised, the hacker cannot hack into other applications.

Access Manager Solution: Modification of Profile Attributes

The security issue labeled "Security Issue: Access to User Profile Attributes" in this document pertains to the threat posed by an untrusted application modifying the profile attributes of the user. The Access Manager solution to this issue does not change the SSO token. The restricted SSO token is identical to the regular SSO token ID. However, the set of Session Service operations that accept restricted SSO token IDs is limited. This functionality enables Access Manager to prevent applications from modifying profile attributes of the user.

Key Aspects of the Access Manager Solution: Cookie Hijacking Security Issues

The following subsections explain some of the key or more complex aspects of the Access Manager solution to the cookie hijacking security issues defined in this document.

The Significance of Creating an Agent Profile for Each Agent

A key task presented in this document is the task of ensuring that each agent has its own agent profile. For Policy Agent 2.2, J2EE agents and web agents differ in regard to the agent profile. Each J2EE agent must have a unique agent profile to function. Web agents, on the other hand, have a default value for the agent profile that all web agent instances can share. Therefore, to configure web agents against session-cookie hijacking, you must perform additional configuration steps after a web agent is installed to ensure that each web agent has a unique agent profile.

Basically, all the configuration steps presented in this document rely on each agent having its own agent profile. As explained in "Access Manager Solution: Shared Session Cookies" on page 9, a situation where applications share session cookies presents a security issue. Access Manager addresses this potential security risk by issuing a unique SSO token to each agent. In order to issue a unique SSO token to each agent, each agent must have its own agent profile.

Access Manager Session Cookies Involved in Issuing Unique SSO Tokens

When Access Manager is configured to issue unique SSO tokens for each Application/Agent, the following cookies are involved:

Cookie Name	Cookie Value (*place holder*)	Example Cookie Domain Information
iPlanetDirectoryPro	*SSO-token*	`amHost.example.com`

The value of this cookie, which is represented in the preceding table with the place holder *SSO-token*, is the actual value of the token. The domain is set to the host name of the Access Manager instance where the user was authenticated.

Cookie Name	Cookie Value (*place holder*)	Example Cookie Domain Information
iPlanetDirectoryPro	*restricted-SSO-token*	`agentHost.example.com`

The value of this cookie, which is represented in the preceding table with the place holder *restricted-SSO-token*, is the actual value of the token. The domain is set to the host name of the agent instance for which the restricted token is issued.

Cookie Name	Example Cookie Value	Example Cookie Domain Information
sunIdentityServerAuthNServer	`https://amHost.example.com:8080`	`.example.com`

The value of this cookie, which is represented in the preceding table with the example URL `https://amHost.example.com:8080`, is the URL of the Access Manager instance where the user was authenticated. The protocol used for this particular example is `HTTPS` while the port number is a non-default example, `8080`. The domain must be set such that it covers all the instances of Access Manager installed on the network.

Enabling Access Manager to Use Unique SSO Tokens

To enable Access Manager to issue unique SSO tokens, you must enable CDSSO. Therefore, though CDSSO is usually enabled for multiple-domain deployments, in this case, CDSSO must be enabled whether the entire deployment is on a single domain or is spread across multiple domains. In no way does enabling CDSSO for a single domain negatively affect the deployment.

The next section describes the steps required to configure Access Manager to prevent session-cookie hijacking from causing a breach of security.

Implementing the Access Manager Solution for Cookie Hijacking Security Issues

The instructions presented in this section provide a solution to the potential risks related to session-cookie hijacking as outlined in this document. This section includes two subsections as follows:

- "Creating or Updating an Agent Profile" on page 12
- "Configuring the Access Manager Deployment Against Cookie Hijacking" on page 17

In Policy Agent 2.2, web agents and J2EE agents use agent profiles in a slightly different way. The information provided in this section on the agent profile explains the differences between the two agent types and provides the actual steps necessary to ensure that each agent has its own agent profile.

This section also provides the other configuration steps necessary to guard web resources in an Access Manager deployment against the threat of session-cookie hijacking.

After you perform the tasks presented in this document, Access Manager starts enforcing restrictions on the sessions it creates. The new configuration enables Access Manager to more closely track aspects of each session. At that point, not only does Access Manager record which agent performed the initial redirection for authentication it also tracks the applications to which an SSO token has been issued. Access Manager uses this information to facilitate the processing of each subsequent request and to prevent unauthorized access to protected web resources.

Creating or Updating an Agent Profile

Since J2EE agents and web agents in the Policy Agent 2.2 software set handle the agent profile differently, the configuration steps can differ. The following table provides information about the agent profile in relationship to the two agent types:

Agent Profile Information	Web Agents	J2EE Agents
The agent profile includes a user ID and password that the agent stores in its `AMAgent.properties` **configuration file and uses to log in to Access Manager in order to request services.**	Yes	Yes
The agent has default credentials for the agent profile.	Yes	No

As the preceding table indicates, J2EE agents in the 2.2 release do not provide default credentials for the agent profile. You create an agent profile in Access Manager Console prior to installing the J2EE agent, which allows you to provide information about the agent profile during installation. If you do not create an agent profile, the J2EE agent will not function. Since each successfully installed J2EE agent has a unique agent profile, no further configuration would be required regarding the agent profile credentials.

Web agents, on the other hand, provide default credentials for the agent profile. Be aware that you should not use these web agent default values in an Access Manager deployment configured against cookie hijacking. Instead, create an agent profile for the web agent instance as described in "To Create an Agent Profile" on page 14. For reference purposes, the default credentials for the agent profile of a web agent are provided in the list that follows:

Web agent default agent profile "user ID"

Value. `UrlAccessAgent`

Location. This user ID is stored in the web agent `AMAgent.properties` file as the value assigned to the property `com.sun.am.policy.am.username`.

Web agent default agent profile "password"

Value. The password of the Access Manager internal authentication user. This user is commonly referred to as `amldapuser`.

Location. This password, which is encrypted, is stored in the web agent `AMAgent.properties` file as the value assigned to the property `com.sun.am.policy.am.password`.

Caution – The web agent should not use the default agent profile for an Access Manager deployment configured against cookie hijacking. Furthermore, while the J2EE agent installer updates the agent with the agent profile credentials, the web agent installer does not. Therefore, creating an agent profile before installing the web agent, though useful, is not by itself sufficient. You must also configure the web agent `AMAgent.properties` configuration file by editing the agent profile ID and password as described in "To Update the Agent Profile Credentials in the Web Agent" on page 16.

This document provides guidance for creating or updating agent profiles. However, the steps can vary depending upon the agent type and the status of the agent. While the steps that follow are appropriate for many scenarios, such as for agents that are yet to be installed, the steps are not appropriate for all possible scenarios.

For example, if you are using a J2EE agent that was previously installed, the pre-existing agent profile is appropriate. You would not be required to perform any of the steps presented in this section about creating or updating the agent profile.

However, in terms of a previously installed J2EE agent, if you decide to change the agent profile user ID and password, even though not required, then configuration steps are necessary. Be aware that the steps required in this scenario are not provided in this document. For such a scenario, refer to the Policy Agent 2.2 documentation for information specifically about *updating* the agent profile.

For more information on creating or updating the agent profile, see the documentation for the specific agent you are using. The individual agent guides are listed along with supported server information in the following chapters of the *Sun Java System Access Manager Policy Agent 2.2 User's Guide*:

Web Agents	Chapter 2, "Access Manager Policy Agent 2.2 Web Agents: Compatibility, Supported Servers, and Documentation," in *Sun Java System Access Manager Policy Agent 2.2 User's Guide*
J2EE Agents	Chapter 3, "Access Manager Policy Agent 2.2 J2EE Agents: Compatibility, Supported Servers, and Documentation," in *Sun Java System Access Manager Policy Agent 2.2 User's Guide*

▼ To Create an Agent Profile

This task is appropriate for web agents and J2EE agents. See the preceding Caution for information about when this task might apply.

Agent profiles apply to specific agents but are created and modified in Access Manager Console. Therefore, tasks related to the agent profile are discussed in Access Manager documentation and Policy Agent documentation, as well as in this document.

Perform the following steps using the Access Manager Console. The key steps of this task involve creating an agent ID and an agent password.

1. **With the Access Control tab selected click the name of the realm for which you would like to create an agent profile.**

2. **Select the Subjects tab.**

3. **Select the Agent tab.**

4. **Click New.**

5. **Enter values for the following fields:**

   ID. Enter the name or identity of the agent. This is the agent profile name, which is the name the agent uses to log into Access Manager. Multi-byte names are not accepted.

   Password. Enter the agent password. This password must be different than the LDAP user password used by Access Manager.

   Password (confirm). Confirm the password.

   Device Status. Select the device status of the agent. The default status is Active. If set to Active, the agent will be able to authenticate to and communicate with Access Manager. If set to Inactive, the agent will not be able to authenticate to Access Manager.

6. **Click Create.**

   The list of agents appears.

7. **(Optional) If you desire, add a description to your newly created agent profile:**

   a. **Click the name of your newly created agent profile from the agent list.**

   b. **In the Description field, enter a brief description of the agent.**

      For example, you can enter the agent instance name or the name of the application it is protecting.

   c. **Click Save.**

Next Steps If you performed this task for a web agent, you must also perform the following task about updating the agent profile. If you performed this task for a J2EE agent, you can skip to "Configuring the Access Manager Deployment Against Cookie Hijacking" on page 17.

▼ To Update the Agent Profile Credentials in the Web Agent

This task applies to web agents only, not to J2EE agents. This task description provides alternatives for steps that differ according to platform: UNIX-based systems or Windows systems. For a more thorough explanation about updating the agent profile for a web agent, see the corresponding Policy Agent 2.2 documentation.

1 **Update the following property in the web agent `AMAgent.properties` configuration file:**

`com.sun.am.policy.am.username`

Replace the value of this property with the agent profile ID (name) you just updated in Access Manager Console. Therefore, this property would then appear similarly to the following:

`com.sun.am.policy.am.username =` *agent-profile-ID*

2 **Change directories to the following:**

- **UNIX-based Systems**

  PolicyAgent-base`/bin`

- **Windows Systems**

  PolicyAgent-base`\bin`

3 **Execute the following script in the command line:**

- **UNIX-based Systems**

  `crypt_util` *agent-profile-password*

- **Windows Systems**

  `cryptit` *agent-profile-password*

where *agent-profile-password* represents the agent profile password you updated in Access Manager Console.

4 **Copy the output obtained after issuing the script in the previous step and paste it as the value for the following property:**

`com.sun.am.policy.am.password`

This property would appear similarly to the example that follows, where *encrypted-password-string* is a place holder for the real encrypted password:

`com.sun.am.policy.am.password =` *encrypted-password-string*

5 **Ensure that protected web resources are accessible.**

a. **Restart the web container.**

b. **Attempt to access any resource protected by the web agent.**

If the attempt to access a protected resource results in a redirection to Access Manager, then the preceding steps were executed properly.

Configuring the Access Manager Deployment Against Cookie Hijacking

At this point in the configuration, each agent has its own agent profile. However, Access Manager has not been configured yet to associate an SSO token to a specific agent profile. The steps in this section enable this type of association. Ultimately, the new configuration introduces “restricted tokens” into the Access Manager deployment, guarding against security issues as described in this document.

▼ To Configure the Access Manager Deployment Against Cookie Hijacking

This task description includes configuration information for agents in the Policy Agent 2.2 software set. Perform the task on every agent instance for which you want to enhance security. The best practice is to perform the task on all the agent instances in the Access Manager deployment. As part of the configuration of each agent instance, you must also make specific configurations directly to Access Manager. For this task, be prepared to access the Access Manager Administration Console, the agent `AMAgent.properties` configuration file, the Access Manager `AMConfig.properties` configuration file, and a browser that can access a protected web resource.

1 **Using the Access Manager Administration Console, access the agent profile configuration page.**

For the steps on navigating within the Access Manager Administration Console to the agent profile configuration page, see “To Create an Agent Profile” on page 14.

2 **Add the appropriate value to the field labeled Agent Key Value.**

Set the agent properties with a key/value pair as illustrated in the example that follows. This property is used by Access Manager to retrieve an agent profile from an agent repository for credential assertions about agents. Currently, only one property is valid. All other properties are ignored. Use the following format:

`agentRootURL=`*protocol*`://`*hostname*`:`*port*`/`

The preceding entry must be precise. Be aware that the string "agentRootURL" is case sensitive. Also, the slash following the port number is required.

protocol Represents the protocol used, such as HTTP or HTTPS.

hostname Represents the host name of the machine on which the agent resides. This machine also hosts the resources that the agent protects.

port Represents the port number on which the agent is installed. The agent listens to incoming traffic on this port and, from the port, intercepts all requests to access resources on the host.

The following is an example of how this property could be set:

```
agentRootURL=https://agentHost.example.com:8080/
```

3 **Edit the appropriate (J2EE agent or web agent) AMAgent.properties configuration file as necessary.**

a. **Set the property that enables CDSSO to true as illustrated:**

- For J2EE agents set the following property as indicated:

```
com.sun.identity.agents.config.cdsso.enable = true
```

- For web agents set the following property as indicated:

```
com.sun.am.policy.agents.config.cdsso.enable = true
```

The preceding property setting enables CDSSO, which is required for each agent instance since the agent will use functionality provided by the CDSSO feature.

b. **Set the property that stores the URL users are directed to after they log in successfully in a deployment enabled for CDSSO:**

- For J2EE agents set the corresponding property as suggested by the following example:

```
com.sun.identity.agents.config.cdsso.cdcservlet.url[0] =
https://amHost.example.com:8080/amserver/cdcservlet
```

- For web agents set the corresponding property as suggested by the following example:

```
com.sun.am.policy.agents.config.cdcservlet.url =
https://amHost.example.com:8080/amserver/cdcservlet
```

4 **Restart the container that hosts the agent.**

5. **Edit the Access Manager `AMConfig.properties` configuration file to reflect the required changes.**

   a. **Set the following property to `true` as illustrated:**

   ```
   com.sun.identity.enableUniqueSSOTokenCookie = true
   ```

   b. **Set the following property exactly as it is illustrated:**

   ```
   com.sun.identity.authentication.uniqueCookieName =
   sunIdentityServerAuthNServer
   ```

   c. **Set the following property to a domain such that it covers all the Access Manager instances installed:**

   ```
   com.sun.identity.authentication.uniqueCookieDomain
   ```

   The following example illustrates how this property would be set if the domain name was `example.com`.

   ```
   com.sun.identity.authentication.uniqueCookieDomain = .example.com
   ```

6. **In the Access Manager Administration Console, select the Configuration tab.**

7. **Scroll as needed to the System Properties list and click Platform.**

8. **In the Cookie Domain list, change the cookie domain name.**

   This step enables Access Manager to set host-specific session cookies instead of domain-wide session cookies.

   a. **Ensure that the default domain, such as "example.com," is selected.**

   b. **Click Remove.**

   c. **Enter the name of the machine hosting the Access Manager instance.**

   For example:

   amHost.example.com

   d. **Click Add.**

9. **Ensure that the proper cookies appear in a browser.**

   a. **Use a browser to access a resource that is protected by the agent that you just configured.**

   b. **Check the browser's cookie settings to ensure that the three following cookies appear:**

Cookie Name	Example Cookie Value	Example Cookie Domain Information
iPlanetDirectoryPro	*SSO-token*	`amHost.example.com`
iPlanetDirectoryPro	*restricted-SSO-token*	`agentHost.example.com`
sunIdentityServerAuthNServer	`https://amHost.example.com:8080`	`.example.com`

For more information about the preceding cookies, see "Access Manager Session Cookies Involved in Issuing Unique SSO Tokens" on page 11.