Technical Note: Sun Java System Access Manager Cross-Domain Single Sign-On

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

Part No: 820–1165–10

070625@17466

Technote: Cross-Domain Single Sign-On

This document contains the following sections:

- "Software Versions " on page 3
- "Introduction" on page 3
- "CDSSO Overview" on page 5
- "Configuring CDSSO" on page 10
- "J2EE Agent Sample Use Case Protocol Exchange" on page 12
- "Web Policy Agents Use Case — Protocol Exchange" on page 31
- "Accessing Sun Resources Online" on page 44

Software Versions

This technical note is based on Sun Java System Access Manager 7.0 SP1, Sun Java System J2EE Agent 2.2, and Sun Java System Web Policy Agent 2.2 . However, this document applies to the following product versions until further notice:

- Access Manager 7.0
- J2EE agent 2.2
- Web Policy Agent 2.2

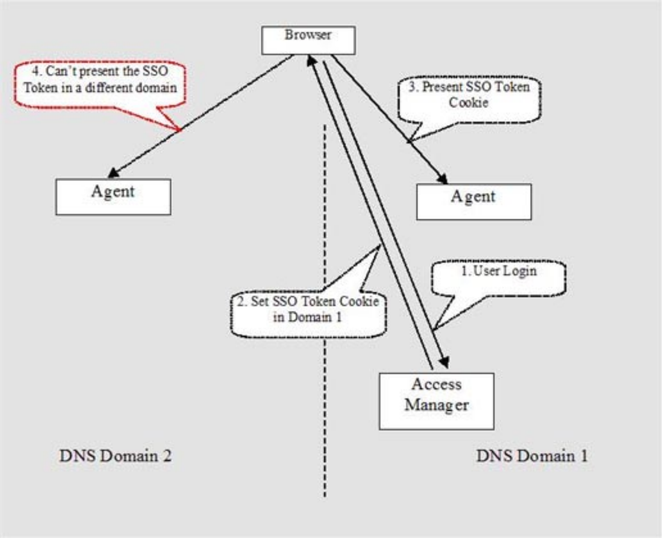
Introduction

The Cross Domain Single Sign-On (CDSSO) is a mechanism by which a Single Sign On (SSO) solution can be extended to Sun Java Enterprise System Access Manager protected resources in different DNS domains. CDSSO makes it possible for users to authenticate once against Access Manager in a primary DNS domain, and then access protected resources in other DNS domains.

CDSSO is a Sun proprietary mechanism from Access Manager, designed before Security Assertion Markup Language (SAML) and the Liberty Alliance Project existed. CDSSO is still available today in Access Manager and it is easier to set up and manage than SAML and Liberty in certain cases.

The CDSSO Challenge

Conventional SSO works via HTTP cookies within a single DNS domain. In such situations, Access Manager and agent-protected resources reside in the same DNS domain. When a user successfully authenticates to Access Manager, an HTTP session cookie (also known as an SSO token) will be set to the user's browser, with Access Manager's DNS domain as the cookie domain. From this point on until the session terminates or expires, the browser will always present the SSO token to any server in the same DNS domain (for example Access Manager agents), based on the HTTP protocol. This allows Access Manager and the policy agents to reexamine the validity of the user session and identity, and then enforce security policies without re-authentication.


Browser
4. Can't present the SSO Token in a different domain
3. Present SSO Token Cookie
Agent
Agent
1. User Login
2. Set SSO Token Cookie in Domain 1
Access Manager
DNS Domain 2
DNS Domain 1

The SSO solution breaks down when the Access Manager and agents reside in different DNS domains. For example, Access Manager and some agents may reside in `www.primary.com` while some other agents reside in `www.partner.com`. During authentication to Access Manager, the SSO token (HTTP cookie) will still be set to the browser with `.primary.com` as the cookie domain. However, when the browser accesses agent-protected resources in `www.partner.com`, it will not present the SSO token to the servers, as dictated by the HTTP protocol. To the Access Manager agents, no SSO token means the user is not authenticated. The agents will force the user to authenticate, which will then fall into a loop, since the Access Manager in the right DNS domain will see the browser does have a valid session SSO token. The Access Manager will redirect the browser back to the original requested resource in `www.partner.com`, thus leads to a non-stopping loop.

CDSSO versus SAML/Liberty

CDSSO has nothing to do with SAML/Liberty even though its implementation uses Liberty-like protocol exchange AuthNResponse. SAML/Liberty solves a broader set of SSO issues where CDSSO focuses on a much narrower subset.

CDSSO requires all Access Manager policy agents to be configured to use a single Access Manager server. This means only one user identity can exist in the entire system. In SAML/Liberty, user identities can exist in multiple systems such as Service Providers (SPs) and Identity Providers (IDPs). SAML/Liberty enables account mapping from IDP to SP. Account mapping from IDP to SP is not possible with CDSSO. Because of the single user store assumption, issues such as account mapping, attribute flow and session synchronization in SAML/Liberty are not relevant to CDSSO. If the situation fits the following, then CDSSO may be a simpler and more suitable solution than SAML/Liberty:

1. Only Sun Java System Access Manager and Sun policy agents are involved.
2. Access Manager policy agents are all configured to use the same Access Manager infrastructure where multiple Access Manager instances can exist.
3. Access Manager uses a single user identity store.
4. Multiple Access Manager instances configured for high-availability must all reside in a single DNS domain. Only policy gents can reside in different DNS domains.

CDSSO Overview

This section describes the over functionality for CDSSO for both J2EE and Web Policy Agents.

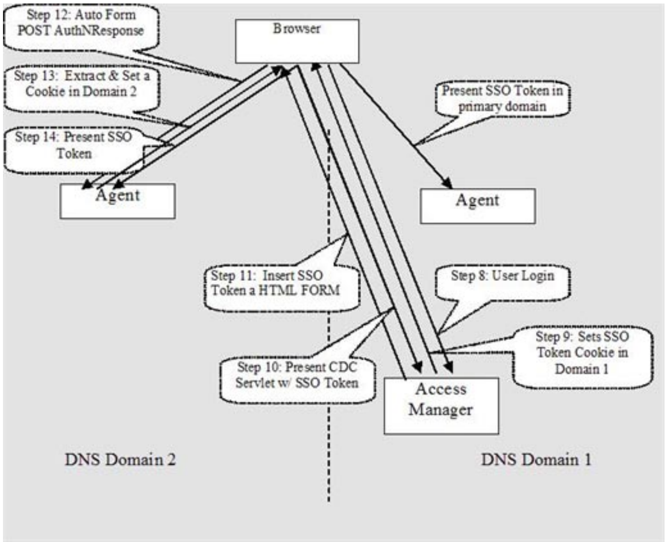
J2EE Agent CDSSO

Based upon the appropriate HTTP protocols, an HTTP cookie will be presented to only a server in the DNS domain that is set in the cookie. A server may only set a cookie within their own domain. Hence, despite having a valid SSO token cookie in one domain, agent protected servers in other domains are never presented with this cookie. CDSSO overcomes the problem with coordinated work between two components:

1. On the Access Manager, Cross-Domain Controller (CDC) serlvet
   `http(s)://am_host:port/amserver/cdcserlvet`
2. On the J2EE agent: CDSSO Redirect Servlet
   `http(s)://agent_host:port/agentapp/sunwCDSSORedirectURI`

The CDSSO Redirect Servlet extracts the SSO Token sent by the CDC Servlet, and then sets the same SSO Token cookie again. This time the SSO Token is set with the agent DNS domain as the cookie domain. The purpose of CDSSO Redirect Servlet is to extract the SSO token sent by CDC Servlet and sets the same SSO token cookie again but with the agent DNS domain as the cookie domain. This process essentially duplicates the SSO token in the agent DNS domain from the Access Manager DNS domain.

In the example illustrated above, the Access Manager server resides in DNS `Domain-1`, and a Policy Agent/Protected Resource resides in DNS `Domain-2`. The following protocol illustrates the CDSSO process:

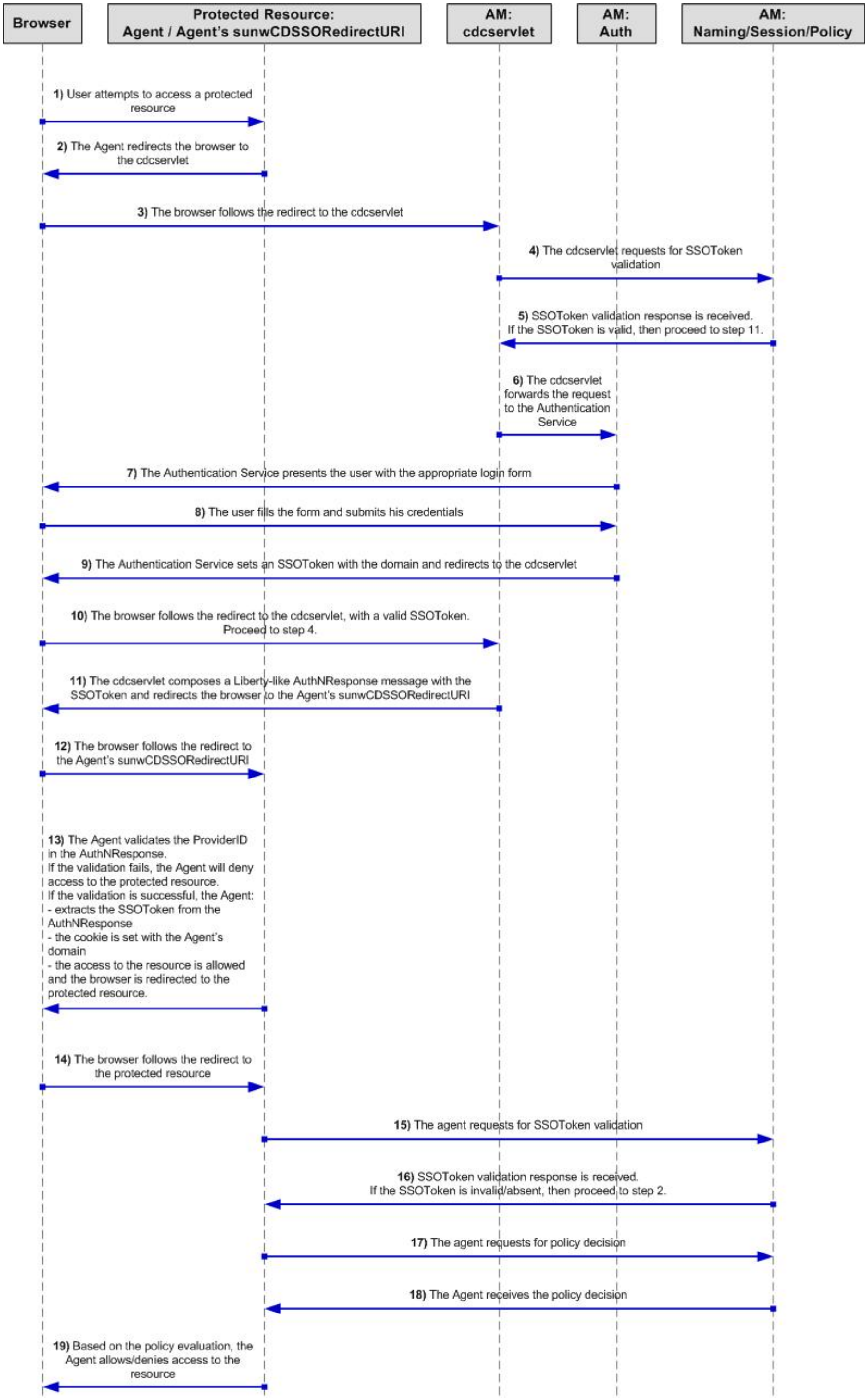

Step 12: Auto Form POST AuthNResponse
Browser
Step 13: Extract & Set a Cookie in Domain 2
Present SSO Token in primary domain
Step 14: Present SSO Token
Agent
Agent
Step 11: Insert SSO Token a HTML FORM
Step 8: User Login
Step 9: Sets SSO Token Cookie in Domain 1
Step 10: Present CDC Servlet w/ SSO Token
Access Manager
DNS Domain 2
DNS Domain 1

1. The user's browser attempts to access an Agent-protected resource in `Domain-2`.
2. With CDSSO enabled on this Agent, the Agent will redirect the browser to the cdcservlet on the Access Manager. Without CDSSO enabled, the Agent will normally redirect the user to Access Manager login URL for user login.
3. The browser follows the redirection and accesses the cdcservlet on the Access Manager.
4. At this time, the cdcservlet will need to determine if the ssotoken is valid and that the cookie is for `Domain-1`. The cdcservlet forwards the request to Access Manager for token validation.
5. Access Manager sends the response to the cdcservlet. If the ssotoken was found to be valid and a cookie present for Domain-1, then proceed to Step 11
6. Since the ssotoken is invalid or absent, the cdcservlet forwards the request to the Access Manager's Authentication Service. When this forwarding of the request is done, it does not involve the client browser. The request is forwarded to the servlet/jsp in the container.
7. The Authentication Service presents the user with the login form.

8. The user provides his credentials on the login page, clicks on the 'Submit' button. The credentials use a POST action to the authentication module.
9. If authentication is successful, the Authentication Service sets the ssotoken cookie for the `Domain-1`. The Authentication Service then redirects the browser back to the cdcservlet.
10. The browser follows the redirection and accesses the cdcservlet on the Access Manager. Now proceed to Step 4.
11. The CDC servlet retrieves the user's SSO Token for Domain-1, composes a Liberty-like AuthNResponse message (LARES) with the SSOToken wrapped inside. The LARES message is contained in a HTML FORM in the HTTP response page. The HTTP response page also contains the directives to automatically post the form without the user interaction to the Agent's `sunwCDSSORedirect` URI.
12. The browser, upon receiving the HTTP response, automatically posts the form to the Agent's `sunwCDSSORedirect` URI.
13. The Agent intercepts this request. Since it is `sunwCDSSORedirect` URI (part of URL that is intercepted by the Agent) the Agent determines that it is a response from cdcservlet and processes the CDSSO response. The Agent validates the ProviderID in the CDSSO response, by comparing it with the registered Providers. If the provider validation fails, the user will be denied access to the protected resource. If the provider validation is successful, the Agent extracts the SSOToken and sets the cookie for Domain-2. Now the Agent retrieves the original requested URL for the protected resource and does a redirect to it.
14. The browser receives the new cookie. Now the browser has two ssotoken cookies, they only differ in their cookie domains. One is for `Domain-1` and another is for Domain-2. The browser follows the redirection to the protected resource, presenting the new SSO token.
15. The Agent intercepts the request for the protected resource, and requests Access Manager to validate the ssotoken.
16. The Access Manager determines if the ssotoken is valid and the cookie is present for `Domain-1`. The Access Manager sends it's response to the Agent. If the ssotoken was evaluated to be invalid or absent, then proceed to Step 2.
17. Since the ssotoken was evaluated to be valid, the Agent next requests Access Manager for the policy decision as pertaining to the protected resource.
18. The Agent receives the policy decision from Access Manager, and evaluates it to determine if the user should be allowed or denied access to the protected resource. Based on the policy evaluation, the Agent enforces the policy.
19. If the policy evaluation resulted in denying access for the user, the user will see a message to that effect in the browser and not be able to access the protected resource. if the policy evaluation resulted in allowing access to the user, the user will be shown the protected resource.

Note – In step 13 above, the current J2EE policy agent implementation sets an SSO token cookie without a domain. Per cookie protocol, this means the cookie will be presented only when accessing the server where the J2EE policy agent reside. The cookie will not be presented to any other servers in the same domain.

The following diagram illustrates the sequence of the steps described in the previous section:



1) User attempts to access a protected resource

2) The Agent redirects the browser to the cdcservlet

3) The browser follows the redirect to the cdcservlet

4) The cdcservlet requests for SSOToken validation

5) SSOToken validation response is received.
If the SSOToken is valid, then proceed to step 11.

6) The cdcservlet forwards the request to the Authentication Service

7) The Authentication Service presents the user with the appropriate login form

8) The user fills the form and submits his credentials

9) The Authentication Service sets an SSOToken with the domain and redirects to the cdcservlet

10) The browser follows the redirect to the cdcservlet, with a valid SSOToken.
Proceed to step 4.

11) The cdcservlet composes a Liberty-like AuthNResponse message with the SSOToken and redirects the browser to the Agent's sunwCDSSORedirectURI

12) The browser follows the redirect to the Agent's sunwCDSSORedirectURI

13) The Agent validates the ProviderID in the AuthNResponse.
If the validation fails, the Agent will deny access to the protected resource.
If the validation is successful, the Agent:
- extracts the SSOToken from the AuthNResponse
- the cookie is set with the Agent's domain
- the access to the resource is allowed and the browser is redirected to the protected resource.

14) The browser follows the redirect to the protected resource

15) The agent requests for SSOToken validation

16) SSOToken validation response is received.
If the SSOToken is invalid/absent, then proceed to step 2.

17) The agent requests for policy decision

18) The Agent receives the policy decision

19) Based on the policy evaluation, the Agent allows/denies access to the resource

Web Policy Agent CDSSO

Web Policy Agent works similarly as the J2EE Policy Agent, except for a slight variance. No CDSSO Redirect Servlet exists on the web policy agent because the agent is a NSAPI plugin. Instead, the web policy agent is invoked via a URL parameter called `sunwMethod`. As a result, the web policy agent combines the above steps 13-16 into a single step without the extra redirection in Step 14. For more details, please refer to the section "Web Policy Agent Sample Protocol Exchange" in this document.

Configuring CDSSO

This section describes the procedures to configure and implement CDSSO.

Configuring Access Manager for CDSSO

The CDC Servlet (`/amserver/cdcservlet`) is always available and enabled on the Access Manager server. There are no special steps to install and configure it. However, if you deploy multiple AM instances behind a load balancer, you need to configure the Access Manager instances accordingly. The configuration has nothing to do with CDSSO. For instructions on how to configure multiple Access Manager instances behind a load balancer, see "Deployment Example: Access Manager Load Balancing, Distributed Authentication, and Session Failover" at the following URL:

http://docs.sun.com/app/docs/doc/819-6258.

The policy agent part of the configuration will vary depending upon whether you are using a single Access Manager instance or multiple Access Manager instances.

Configuring the J2EE Agent for CDSSO

For J2EE agents, the CDSSO Redirect Servlet is also deployed but disabled by default. No special steps are required during agent installation. After the installation, you should see the following default CDSSO processing properties in the `AMAgent.properties` file:

```
com.sun.identity.agents.config.cdsso.enable = false
com.sun.identity.agents.config.cdsso.redirect.uri = /agentapp/sunwCDSSORedirectURI
com.sun.identity.agents.config.cdsso.cdcservlet.url[0] =
https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
com.sun.identity.agents.config.cdsso.clock.skew = 0
com.sun.identity.agents.config.cdsso.trusted.id.provider[0] =
     https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
```

Note – the URL `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet` points to the CDC servlet on the Access Manager. The Access Manager protocol, host and port are provided during the agent installation. The two properties, `cdcservlet.url` and `provider`, are typically added by the agent installer.

If a J2EE agent resides in the same DNS domain as the Access Manager, CDSSO is not necessary and should be disabled. Check `AMAgent.properties` and ensure this property value remains the default "false": `com.sun.identity.agents.config.cdsso.enable = false`.

If a J2EE agent resides in a different DNS domain than the Access Manager, CDSSO can be enabled like this:

```
com.sun.identity.agents.config.cdsso.enable = true
com.sun.identity.agents.config.cdsso.redirect.uri = /agentapp/sunwCDSSORedirectURI
com.sun.identity.agents.config.cdsso.cdcservlet.url[0] =
https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
com.sun.identity.agents.config.cdsso.clock.skew = 0
com.sun.identity.agents.config.cdsso.trusted.id.provider[0] =
      https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
```

If multiple Access Manager instances are behind a load balancer, then the actual Access Manager CDC servlet URL for all Access Manager instances should be added to the trusted ID provider list. This list will be consulted when the policy agent CDSSO Redirect Servlet receives the AuthNResponse. If the individual Access Manager CDC servlet URLs are not in the list, the agent rejects the AuthnReponse from a non trusted provider like this:

```
ERROR: LibertyAuthnResponseHandler : Response received from an untrusted provider
       - https://ide-14.red.iplanet.com:443/amserver/cdcservlet
```

For example: if `ide-14` and `ide-15` are two SSL-enabled Access Manager instances behind a load balancer `am-pool0.red.iplanet.com:8443`, then the agent CDSSO configuration should look like this:

```
com.sun.identity.agents.config.cdsso.enable = true
com.sun.identity.agents.config.cdsso.redirect.uri = /agentapp/sunwCDSSORedirectURI
com.sun.identity.agents.config.cdsso.cdcservlet.url[0] =
https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
com.sun.identity.agents.config.cdsso.clock.skew = 0
com.sun.identity.agents.config.cdsso.trusted.id.provider[0] =
https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
com.sun.identity.agents.config.cdsso.trusted.id.provider[1] =
https://ide-14.red.iplanet.com:443/amserver/cdcservlet
com.sun.identity.agents.config.cdsso.trusted.id.provider[2] =
https://ide-15.red.iplanet.com:443/amserver/cdcservlet
```

Technically, you don't need the load balancer's URL in the trusted provider list. The load balancer URL is derived from the detected Access Manager host, port, and URL, and then added by the agent installer. But it doesn't hurt to have the load balancer URL in the trusted provider list.

The clock skew factor is used to handle minor system clock drifts between the agent and the Access Manager instances. Even with the presence of this parameter, it's a good practice to synchronize system clocks between Access Manager servers and policy agents with a NTP service.

Configuring the Web Policy Agent for CDSSO

Web policy agents CDSSO has considerably fewer configurable parameters than J2EE policy agents. During the installation of the web policy agents, you are prompted to enable or disable CDSSO. If you choose to enable, after the installation, you should see the following default CDSSO processing properties in the `AMAgent.properties` file:

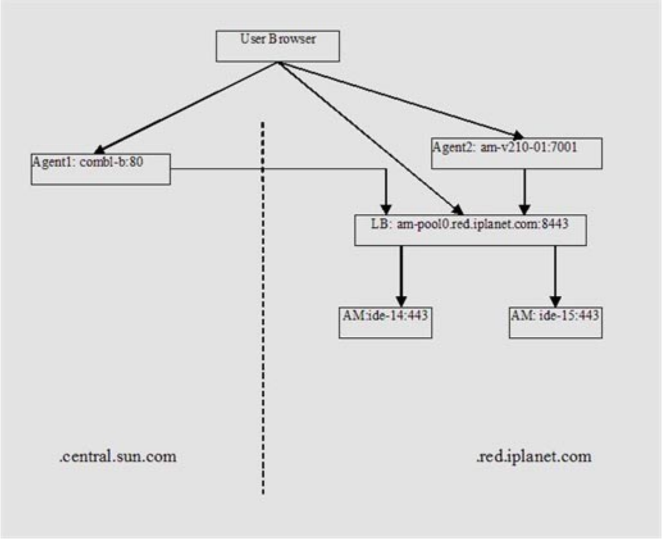
```
com.sun.am.policy.agents.config.cdsso.enable=true
com.sun.am.policy.agents.config.cdcservlet.url =
https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet
```

The URL `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet` points to the CDC servlet on the Access Manager. The Access Manager protocol, host and port are provided during the agent installation. The property `cdcservlet.url` is typically added by the agent installer.

J2EE Agent Sample Use Case Protocol Exchange

The following are actual protocol exchanges in the two use cases. In both use cases, the configuration is as follows:

- The primary domain (where Access Manager resides) is `.iplanet.com`. The non-primary domain is `.sun.com`.
- In the primary domain, there are two Access Manager instances: `ide-14.red.iplanet.com:443` and `ide-15.red.iplanet.com:443`. Both are behind a load balancer `am-pool0.red.iplanet.com:8443`.
- In the primary domain, Agent #2 resides in `am-v210-01.red.iplanet.com:7001` with CDSSO disabled because it's in the same DNS domain as the Access Manager instances.
- In the non-primary domain, Agent #1 resides in `comal-b.central.sun.com:80` with CDSSO enabled.
- A protected resource `/app1/test1.html` is deployed on the servers of both agents.



In the use cases, we will demonstrate a CDSSO sequence from the primary domain to the non-primary domain, and the reverse.

J2EE Use Case 1: Accessing a Protected Resource in the Primary Domain First

In this use case, an unauthenticated user first accesses a resource under the agent #2 in the Access Manager DNS domain (the primary domain). After the authentication, the Access Manager sets an SSO token in domain `.iplanet.com`. Then the user accesses another resource under agent #1 in a different domain `.central.sun.com`. The CDSSO sequence will be invoked and access will be allowed without re-authentication.

1. An unauthenticated user attempts to access `http://am-v210-01.red.iplanet.com:7001/app1/test1.html`. The agent intercepts the request and receives no SSO token. The agent responds with a redirection to the Access Manager login page.

   REQUEST:

   ```
   GET /app1/test1.html HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   If-Modified-Since: Tue, 20 Jun 2006 11:03:04 GMT
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   ```

```
SV1; .NET CLR 1.1.4322)
Host: am-v210-01.red.iplanet.com:7001
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Date: Wed, 02 Aug 2006 12:26:47 GMT
Location: https://am-pool0.red.iplanet.com:8443/amserver/UI/Login?goto=
        http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001%2Fapp1%2Ftest1.html
Content-Type: text/html
Connection: Close
```

2. The browser follows the redirection to access the Access Manager login page.

   REQUEST:

   ```
   GET /amserver/UI/Login?goto=
   http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001%2Fapp1%2Ftest1.html HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash,application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   If-Modified-Since: Tue, 20 Jun 2006 11:03:04 GMT
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-pool0.red.iplanet.com:8443
   Connection: Keep-Alive
   ```

   RESPONSE:

   ```
   HTTP/1.1 200 OK
   Server: Sun-ONE-Web-Server/6.1
   Date: Wed, 02 Aug 2006 12:26:52 GMT
   Content-type: text/html;charset=UTF-8
   Cache-control: private
   Pragma: no-cache
   Expires: 0
   X-dsameversion: 7 2005Q4
   Am_client_type: genericHTML
   Set-cookie: JSESSIONID=54C2BEA3AB9BEE7AC172AD396F6C012A;Path=/;Secure
   Set-cookie: AMAuthCookie=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoBDs4FiDT7O
   zpFDjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
         ;Domain=.iplanet.com;Path=/
   Set-cookie: amservercookie=0C;Domain=.iplanet.com;Path=/

   <.... login in page content omitted by the author >
   ```

3. The user types in his credential on the login page and clicks Submit. A login form is posted to Access Manager. If the user authenticates successfully, the Access Manager responds by setting an SSO token (`iPlanetDirectoryPro`) in the domain `.iplanet.com`. The response

also redirects the browser to the original requested resource `http://am-v210-01.red.iplanet.com:7001/app1/test1.html`.

REQUEST:

```
POST /amserver/UI/Login HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/msword, */*
Referer: https://am-pool0.red.iplanet.com:8443/amserver/UI/
      Login?goto=http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001%2Fapp1%2Ftest1.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Content-Length: 144
Cache-Control: no-cache
Cookie: JSESSIONID=54C2BEA3AB9BEE7AC172AD396F6C012A;
       AMAuthCookie=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoBDs4FiDT7OzpF
       DjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
       amservercookie=0C
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-ONE-Web-Server/6.1
Date: Wed, 02 Aug 2006 12:27:01 GMT
Content-length: 0
Content-type: text/html
Cache-control: private
Pragma: no-cache
X-dsameversion: 7 2005Q4
Am_client_type: genericHTML
X-autherrorcode: 0
Location: http://am-v210-01.red.iplanet.com:7001/app1/test1.html
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2
    FKcoBDs4FiDT7OzpFDjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
Domain=.iplanet.com;Path=/
Set-cookie: AMAuthCookie=LOGOUT;Domain=.iplanet.com;Expires=Thu,
01-Jan-1970 00:00:10 GMT;Path=/
Connection: close
```

4. The browser follows the redirection to access `http://am-v210-01.red.iplanet.com:7001/app1/test.html`. Note the SSO token cookie `iPlanetDirectoryPro` is sent in the HTTP request to the server. The agent validates

the SSO token and evaluates policies by interacting with the Access Manager in the background. If the access is allowed, the server responds with the content of the protected resource.

REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/msword, */*
Pragma: no-cache
Accept-Language: en-us
Cookie: amservercookie=0C; iPlanetDirectoryPro=AQIC5wM2LY4Sfcw%
   2F71xSeh8udj3%2FKcoBDs4FiDT7OzpFDjQ%3D%40
   AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-v210-01.red.iplanet.com:7001
Cache-Control: no-cache
```

RESPONSE:

```
HTTP/1.1 200 OK
Date: Wed, 02 Aug 2006 12:27:02 GMT
Content-Length: 88
Content-Type: text/html
Last-Modified: Tue, 20 Jun 2006 11:03:04 GMT
Set-Cookie: JSESSIONID=GQhWKgrXz1R8jCSpgnc1jXtzMd0M
jwn1y9NXPjpZGCQn7jhX5wKd!384704559; path=/
Accept-Ranges: bytes
Connection: Close
<html>
<head>
<title>Test1 HTMOL</title>
</head>
TEST1 HTML
</body>
</html>
```

5. The user now attempts to access another resource `http://comal-b.central.sun.com:80/app1/test1.html`. Note the SSO token `iPlanetDirectoryPro` is not sent in the HTTP request because the server `comal-b.central.sun.com` does not match the cookie domain `.iplanet.com`. The agent, receiving no SSO token, responds by redirecting the browser to the CDC servlet URL `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet`.

   REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Cookie: SUN_ID=69.196.39.237:227251153914164
If-Modified-Since: Wed, 19 Jul 2006 14:43:46 GMT
If-None-Match: W/"88-1153320226000"
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-Java-System/Application-Server
Date: Wed, 02 Aug 2006 12:27:09 GMT
Content-type: text/html
X-powered-by: Servlet/2.4
Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?goto=
   http%3A%2F%2Fcomal-b.central.sun.com
   %3A80%2Fagentapp%2FsunwCDSSORedirectURI&refererservlet=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%
   2FsunwCDSSORedirectURI&MajorVersion=1&MinorVersion=0&RequestID;
   =s8c70ff292d4b9f9fbb211003528b7ab90de41229&ProviderID;=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F%3FRealm%3D%252F&IssueInstant;
   =2006-08-02T12%3A27%3A09Z&ForceAuthn;=false&IsPassive;=false&Federate;=false
Set-cookie: amFilterCDSSORequest=AQICAtwmVLBfMe/
PgTWWJqWPSfO2eZo6rYLpQLiSI2Uk+Es+I25/
7Pb5lDpLfNbM1S64amLqY9RLg1gib2HzbGqM+GKp/aF/
PslJYgcOwjKzAjZCBX+fDUtjQazNCAD+XwOdOnVsdKuGHNs=; Path=/
Connection: close
```

A cookie `amFilterCDSSORequest` is set by the agent to save the user requested URL, its access type (GET/POST), etc. and `AuthnRequestID` (value of RequestID query parameter). This cookie is set before redirecting to the Access Manager's CDC Servlet. After getting the `AuthnResponse` later from the CDC Servlet, the Agent then retrieves the information from the `amFilterCDSSORequest` cookie to continue with the User's Original requested URL.

The redirection URL contains some parameters to be carried to the CDC servlet. Some of these parameters are:

goto — The URL to which CDC servlet will forward AuthNResponse.

MajorVersion — Major version is set 1. It is Liberty Federation Protocol major version.

MinorVersion — The minor version is set to 1. It is Liberty Federation Protocol minor version.

RequestID — Is an Authn Request ID. It is a uniquely generated id. It is of the form `sfollowed-by-20-digit-hexadecimal-string.` This is sent to CDC

Servlet so that the its AuthnResponse later can contain this unique identifier. The RequestID is used to tie the response coming back. It is verified when the response comes back from the CDC servlet.

ProviderID It is Service Provider ID, which is the agent. The value will be of the form: `http://agent-host:port/?Realm=RealmName`. Where RealmName is what is configured for property `com.sun.identity.agents.config.organization.name` in `AMAgent.properties`.

IssueInstant It is the time at which the AuthnRequest was created (being sent), in UTC format.

6. The browser follows the redirection to access the CDC servlet. Note the SSO token `iPlanetDirectoryPro` is sent in the HTTP request because the server DNS domain matches the cookie domain. The CDC servlet validates the SSO token and responds with an HTML page. The page contains an HTML FORM which will be automatically posted to CDSSO Redirect URL on the agent (`http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI`, based on the "goto" parameter earlier). The form's hidden field LARES is an encoded Liberty-like AuthnResponse that contains the existing SSO Token in the domain `.iplanet.com`.

   REQUEST:

   ```
   GET /amserver/cdcservlet?goto=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F
   agentapp%2FsunwCDSSORedirectURI&refererservlet;=http%3A%2F%2
      Fcomal-b.central.sun.com%3A80%2Fagentapp%2FsunwCDSSORedirectURI&
      MajorVersion=1&MinorVersion=0&RequestID;
      =s8c70ff292d4b9f9fbb211003528b7ab90de41229&ProviderID;
      =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F%3FRealm%3D%252F&IssueInstant;
      =2006-08-02T12%3A27%3A09Z&ForceAuthn;=false&IsPassive;
      =false&Federate;=false HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
        application/x-shockwave-flash,application/vnd.ms-excel,
        application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   Cookie: JSESSIONID=54C2BEA3AB9BEE7AC172AD396F6C012A; amservercookie=0C;
      iPlanetDirectoryPro=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoBDs4FiDT7OzpF
      DjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23
   If-Modified-Since: Wed, 19 Jul 2006 14:43:46 GMT
   If-None-Match: W/"88-1153320226000"
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-pool0.red.iplanet.com:8443
   Connection: Keep-Alive
   ```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Wed, 02 Aug 2006 12:27:10 GMT
Content-type: text/html
Pragma: no-cache
Content-length: 3788
Connection: keep-alive

<HTML>
<BODY Onload="document.Response.submit()">
<FORM NAME="Response" METHOD="POST" ACTION=
"http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI">
<INPUT TYPE="HIDDEN" NAME="LARES" VALUE="PGxpYjpBdXRoblJlc3BvbnNlIHhtbG
5zOmxpYj0iaHR0cDovL3Byb2plY3RsaWJlcnR5Lm9yZy9zY2hlbWFzL2NvcmUvMjAwMi8xM
IgeG1sbnM6c2FtbD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6MS4wOmFzc2VydGlvbiIge
G1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjEuMDpwcm90b2NvbCIgeG1s
...
bnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiIHhtbG5zOnhzaT0
iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiIFJlc3BvbnNlSU
0ZmIxODg2Nzc4ZTBkMzMyMmEzMzFhYTg4MzMzOTMxNjZmMmYwIiAgSW5SZXNwb25zZVRvPS
JzOGM3MGZmMjkyZDRiOWY5ZmJiMjExMDAzNTI4YjdhYjkwZGU0MTIyOSIgIE1ham9yVmVyc
2lvbj0iMSIgIE1pbm9yVmVyc2lvbj0iMCIgIElzc3VlSW5zdGFudD0iMjAwNi0wOC0wMlQx
g=="/>
</FORM>
</BODY></HTML>
```

The decoded AuthnResponse (line-wrapped) looks like this:

```
<lib:AuthnResponse xmlns:lib=
"http://projectliberty.org/schemas/core/2002/12" xmlns
:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:samlp=
"urn:oasis:names:tc:SAML:1.0:protocol" xmlns:ds=
"http://www.w3.org/2000/09/xmldsig<" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
ResponseID="sb3f4fb1886778e0d3322a331aa8833393166f2f0"
InResponseTo="s8c70ff292d4b9f9fbb211003528b7ab90de41229"
MajorVersion="1"
MinorVersion="0"
IssueInstant="2006-08-02T12:27:10Z">
<samlp:Status>
<samlp:StatusCode Value="samlp:Success">
</samlp:StatusCode>
</samlp:Status>
<saml:Assertion  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xmlns:lib="http://projectliberty.org/schemas/core/2002/12"
id="s16703a57e86c8163160b2a2ab3ce76111cc9ed5a01"
```

```
MajorVersion="1"
MinorVersion="0"
AssertionID="s16703a57e86c8163160b2a2ab3ce76111cc9ed5a01"
Issuer="https://ide-14.red.iplanet.com:443/amserver/cdcservlet"
IssueInstant="2006-08-02T12:27:10Z"
InResponseTo="s8c70ff292d4b9f9fbb211003528b7ab90de41229"
xsi:type="lib:AssertionType">
<saml:Conditions  NotBefore="2006-08-02T12:27:10Z"
NotOnOrAfter="2006-08-02T12:28:10Z" >
<saml:AudienceRestrictionCondition>
<saml:Audience>http://comal-b.central.sun.com:80/?Realm=%2F</saml:Audience>
</saml:AudienceRestrictionCondition>
</saml:Conditions>
<saml:AuthenticationStatement
AuthenticationMethod="LDAP"
AuthenticationInstant="2
006-08-02T12:27:01Z"
ReauthenticateOnOrAfter="2006-08-02T12:28:10Z"
xsi:type="lib:AuthenticationStatementType">
<saml:Subject
xsi:type="lib:SubjectType"><saml:NameIdentifier
NameQualifier="https://ide-14.red.iplanet.com:443/amserver/cdcservlet">
AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoBDs4FiDT7OzpFDjQ%3D%40AAJTSQACMTEAAl
MxAAIwMQ%3D%3D%23</saml:NameIdentifier>
<saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer
</saml:ConfirmationMe>
</saml:SubjectConfirmation>
<lib:IDPProvidedNameIdentifier
NameQualifier="https://ide-14.red.iplanet.com:443/amserver/cdcservlet" >
AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoBDs4FiDT7OzpFDjQ%3D%40AAJTS
QACMTEAAlMxAAIwMQ%3D%3D%23</lib:IDPProvidedNameIdentifier>
</saml:Subject><saml:SubjectLocality  IPAddress="192.18.72.87"
DNSAddress="ide-14.red.iplanet.com" />
<lib:AuthnContext><lib:AuthnContextClassRef>http://www.projectliberty.org/
schemas/authctx/classes/Password</lib:AuthnContextClassRef>
<lib:AuthnContextStatementRef>http://www.projectliberty.org/schemas/
authctx/classes/Password</lib:></lib:AuthnContext>
</saml:AuthenticationStatement></saml:A>
<lib:ProviderID>https://ide-14.red.iplanet.com:443/amserver/cdcservlet
</lib:Provide></lib:AuthnResponse>
```

7. The browser automatically posts the form with LARES to `http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI` without the user interaction. The agent responds by setting a new SSO token `iPlanetDirectoryPro` with an empty cookie domain. A cookie without a domain will be restricted to be sent to the originating server only in the future. Also note the cookie value is exactly the same as the one

set in Step 3 response by Access Manager. The HTTP response also redirects the browser to the original requested resource `http://comal-b.central.sun.com:80/app1/test1.html`.

REQUEST:

```
POST /agentapp/sunwCDSSORedirectURI HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Content-Length: 3592
Cookie: amFilterCDSSORequest=AQICAtwmVLBfMe/PgTWWJqWPSfO2eZo6rYLpQLiSI2Uk+Es+I25/
7Pb5lDpLfNbM1S64amLqY9RLg1gib
   2HzbGqM+GKp/aF/PslJYgcOwjKzAjZCBX+fDUtjQazNCAD+XwOdOnVsdKuGHNs=;
   SUN_ID=69.196.39.237:227251153914164
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-Java-System/Application-Server
Date: Wed, 02 Aug 2006 12:27:12 GMT
Content-type: text/html
X-powered-by: Servlet/2.4
Location: http://comal-b.central.sun.com:80/app1/test1.html
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%
   2FKcoBDs4FiDT7OzpFDjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23; Path=/
Set-cookie: amFilterCDSSORequest=reset;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
Connection: close
```

In responding to this request, the agent goes through the following steps to validate the received AuthnResponse:

a. First the requestID (saved in the `amFilterCDSSORequest` cookie) is verified against the responseID of the AuthnResponse.

b. The status code of the AuthnResponse is verified to see if it is successful.

c. The assertions are extracted from the AuthnResponse. There should be only 1.

d. From the Assertion, the issuer is extracted and is verified against the policy agent list of trusted ID providers. If the issuer is not in the policy agent trusted list, then user request is blocked. These trusted ID providers are governed by property, as we discussed in the configuration section, `com.sun.identity.agents.config.cdsso.trusted.id.providerx`. These IDs should contain URLs of the actual Access Manager instances (not the load-balancer URL).

e. The conditions that are in the assertion are also validated. The main one is the date validity condition. The date validity attributes, not before and notOnorAfter, are verified to verify the assertion has not expired. Hence time synchronization between Access Manager and Agent is essential. Also the skew factor provided in AMAgent `com.sun.identity.agents.config.cdsso.clock.skew` helps to overcome any network latencies.

In the response, cookie `amFilterCDSSORequest` is removed by setting the expiration date in the past.

8. The browser follows the redirection to access the protected resoruce again at `http://comal-b.central.sun.com:80/app1/test.html`. Note the new SSO token is sent to the server. The agent validates the SSO token, evaluates the policies and allows the access. The server responds with the content of the protected resource.

   REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
    application/x-shockwave-flash, application/vnd.ms-excel,
    application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Cookie: iPlanetDirectoryPro=AQIC5wM2LY4Sfcw%2F71xSeh8udj3%2FKcoB
   Ds4FiDT7OzpFDjQ%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
     SUN_ID=69.196.39.237:227251153914164
```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-Java-System/Application-Server
Date: Wed, 02 Aug 2006 12:27:13 GMT
Content-length: 88
Content-type: text/html
X-powered-by: Servlet/2.4
Etag: W/"88-1153320226000"
Last-modified: Wed, 19 Jul 2006 14:43:46 GMT
Connection: close

<html>
<head>
<title>Test1 HTML</title>
```

```
</head>
body>
Test1 HTML
</body>
</html>
```

J2EE Agent Use Case 2: Accessing a Protected Resource in a Non-Primary Domain First

In this use case, an unauthenticated user first accesses a protected resource in the non-primary domain (.sun.com). He then accesses a protected resource in the primary domain (.iplanet.com).

1. An unauthenticated user attempts to access http://comal-b.central.sun.com:80/app1/test1.html. The policy agent intercepts the request and receives no SSO token. Because the SSO is enabled, the agent responds with a redirection to the Access Manager CDC servlet URL https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet.

   REQUEST:

   ```
   GET /app1/test1.html HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   Cookie: SUN_ID=69.196.39.237:227251153914164
   If-Modified-Since: Wed, 19 Jul 2006 14:43:46 GMT
   If-None-Match: W/"88-1153320226000"
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: comal-b.central.sun.com
   ```

   RESPONSE:

   ```
   HTTP/1.1 302 Moved Temporarily
   Server: Sun-Java-System/Application-Server
   Date: Tue, 01 Aug 2006 17:43:58 GMT
   Content-type: text/html
   X-powered-by: Servlet/2.4
   Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?goto=
     http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%
     2FsunwCDSSORedirectURI&refererservlet;
     =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F
      agentapp%2FsunwCDSSORedirectURI&MajorVersion=1&MinorVersion=0&RequestID;
   ```

```
   =sa51a95ae420a2a8bb2d608740680c9df6e767dc3&ProviderID ;
   =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F%3FRealm%3D%252F&IssueInstant;
   =2006-08-01T17%3A43%3A58Z&ForceAuthn;=false&IsPassive;=false&Federate;=false
Set-cookie: amFilterCDSSORequest=AQICAtwmVLBfMe/
   PgTWWJqWPSfO2eZo6rYLpQLiSI2Uk+Es+I25/7Pb5lDpLfNbM1S64amLqY9RLg1i9nEXzWfcn
   BEVZS5SdG2pJtTdMzEgo/o/MARoPq//EMt766UEXFT6aOUAtME0or70=; Path=/
Connection: close
```

2. The browser follows the redirection to access the CDC servlet without any SSO token. The CDC servlet responds with a login page.

   REQUEST:

```
GET /amserver/cdcservlet?goto=http%3A%2F%2
   Fcomal-b.central.sun.com%3A80%2Fagentapp%2FsunCDSSORedirectURI
   &refererservlet;=http%3A%2F%2Fcomal-b.central.sun.com%3A80
   %2Fagentapp%2FsunwCDSSORedirectURI&MajorVersion=1
   &MinorVersion=0&RequestID;=sa51a95ae420a2a8bb2d608740680c9df6e767dc3&
   ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F%3FRealm%3D
   %252F&IssueInstant;=2006-08-01T17%3A43%3A58Z
   &ForceAuthn;=false&IsPassive;=false&Federate;=false HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
If-Modified-Since: Wed, 19 Jul 2006 14:43:46 GMT
If-None-Match: W/"88-1153320226000"
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Connection: Keep-Alive
```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Tue, 01 Aug 2006 17:44:02 GMT
Content-type: text/html;charset=UTF-8
Cache-control: private
Pragma: no-cache
Expires: 0
X-dsameversion: 7 2005Q4
Am_client_type: genericHTML
Set-cookie: JSESSIONID=B38B1B717BDD9EE781995CCEC058A70D;Path=/;Secure
Set-cookie: AMAuthCookie=AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7EGE4o
JXk%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
   Domain=.iplanet.com;Path=/
Set-cookie: amservercookie=0C;Domain=.iplanet.com;Path=/
```

```
<.... login page content omitted by author ...>
```

3. The user types in his credential on the login page and clicks Submit. A login form is posted to Access Manager. If the user authenticates successfully, the Access Manager responds by setting an SSO token (iPlanetDirectoryPro) in the domain `.iplanet.com`. The response also redirects the browser back to the CDC servlet `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet`.

   REQUEST:

   ```
   POST /amserver/UI/Login HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Referer: https://am-pool0.red.iplanet.com:8443/amserver/c
      dcservlet?goto=http%3A%2F%2Fcomal-b.central.sun.com%3A80
      %2Fagentapp%2FsunwCDSSORedirectURI&refererservlet;
      =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%2
      FsunwCDSSORedirectURI&MajorVersion=1&MinorVersion=0&RequestID;
      =sa51a95ae420a2a8bb2d608740680c9df6e767dc3
      &ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com
      %3A80%2F%3FRealm%3D%252F&IssueInstant;
      =2006-08-01T17%3A43%3A58Z
      &ForceAuthn;=false&IsPassive;=false&Federate;=false
   Accept-Language: en-us
   Content-Type: application/x-www-form-urlencoded
   Connection: Keep-Alive
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-pool0.red.iplanet.com:8443
   Content-Length: 600
   Cache-Control: no-cache
   Cookie: JSESSIONID=B38B1B717BDD9EE781995CCEC058A70D;
   AMAuthCookie=AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7EGE43D
      %40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23; amservercookie=0C
   ```

   RESPONSE:

   ```
   HTTP/1.1 302 Moved Temporarily
   Server: Sun-ONE-Web-Server/6.1
   Date: Tue, 01 Aug 2006 17:44:15 GMT
   Content-length: 0
   Content-type: text/html
   Cache-control: private
   Pragma: no-cache
   X-dsameversion: 7 2005Q4
   Am_client_type: genericHTML
   ```

```
X-autherrorcode: 0
Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?
   TARGET=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F
   agentapp%2FsunwCDSSORedirectURI&refererservlet;=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%2F
   sunwCDSSORedirectURI&MajorVersion=1&MinorVersion=0&RequestID;
   =sa51a95ae420a2a8bb2d608740680c9df6e767dc3
   &ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com
   %3A80%2F%3FRealm%3D%252F&IssueInstant;
   =2006-08-01T17%3A43%3A58Z&ForceAuthn;
   =false&IsPassive;=false&Federate;=false
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7
   EGE4oJXk%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
   Domain=.iplanet.com;Path=/
Set-cookie: AMAuthCookie=LOGOUT;Domain=.iplanet.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT;Path=/
Connection: close
```

4. The browser follows the redirection to access the CDC servlet again. This time the SSO token `iPlanetDirectoryPro` is sent in the HTTP request because the server DNS domain matches the cookie domain. The CDC servlet validates the SSO token and responds with an HTML page. The page contains a HTML FORM which will be automatically posted to CDSSO Redirect URL on the agent (`http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI`). The form's hidden field LARES is an encoded Liberty-like AuthnResponse that contains the existing SSO Token in the domain `.iplanet.com`.

   REQUEST:

```
GET /amserver/cdcservlet?TARGET=
http%3A%2F%2Fcomal-b.central.sun.com
   %3A80%2Fagentapp%2FsunwCDSSORedirectURI&
   refererservlet;=http%3A%2F%2Fcomal-b.central.sun.com
   %3A80%2Fagentapp%2FsunwCDSSORedirectURI&MajorVersion=1
   &MinorVersion=0&RequestID;=sa51a95ae420a2a8bb2d608740680c9df6e767dc3&ProviderID;
   =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2F%3FRealm%3D%252F&IssueInstant;
   =2006-08-01T17%3A43%3A58Z&ForceAuthn;=false&IsPassive;=false&Federate;
   =false HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Referer: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?
   goto=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%
   2FsunwCDSSORedirectURI&refererservlet;
   =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fagentapp%
   2FsunwCDSSORedirectURI&MajorVersion=1&MinorVersion=0
   &RequestID;=sa51a95ae420a2a8bb2d608740680c9df6e767dc3&
   ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3
   A80%2F%3FRealm%3D%252F&IssueInstant;=2006-08-01T17%3A43%3A58Z&
   ForceAuthn;=false&IsPassive;=false&Federate;=false
Accept-Language: en-us
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Cache-Control: no-cache
Cookie: JSESSIONID=B38B1B717BDD9EE781995CCEC058A70D; amservercookie=0C;
   iPlanetDirectoryPro=AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7EGE4
oJXk%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23
```

RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Tue, 01 Aug 2006 17:44:16 GMT
Content-type: text/html
Pragma: no-cache
Content-length: 3776
Connection: keep-alive

<HTML>
<BODY Onload="document.Response.submit()">
<FORM NAME="Response" METHOD="POST"
ACTION="http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI">
<INPUT TYPE="HIDDEN" NAME="LARES" VALUE="PGxpYjpBdXRoblJlc3BvbnNlIHhtbG5z
OmxpYj0iaHR0cDovL3Byb2plY3RsaWJlcnR5Lm9yZy9zY2hlbWFzL2NvcmUvMjAwMi8xMiIge
G1sbnM6c2FtbD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6MS4wOmFzc2VydGlvbiIgeG1sbn
M6c2FtbHA9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjEuMDpwcm90b2NvbCIgeG1sbnM6ZHM
9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiIHhtbG5zOnhzaT0iaHR0cDov
L3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiIFJlc3BvbnNlSUQ9InM4MjMyN
...
Tg3ODBhNDYwNTFkZTRlNjQzZDZhNmQ4NDQ3OWRiMTBkYmFiIiAgSW5SZXNwb25zZVRvPSJzYT
xYTk1YWU0MjBhMmE4YmIyZDYwODc0MDY4MGM5ZGY2ZTc2N2RjMyIgIE1ham9yVmVyc2lvbj0i
MSIgIE1pbm9yVmVyc2lvbj0iMCIgIElzc3VlSW5zdGFudD0iMjAwNi0wOC0wMVQxNzo0NDoxN
loiPjxzYW1scDpTdGF0dXM+CjxzYW1scDpTdGF0dXNDb2RlIFZhbHVlPSJzYW1scDpTdWNjZX
NzIj4KPC9zYW1scDpTdGF0dXNDb2RlPgo8L3NhbWxwOlN0YXR1cz4KPHNhbWw6QXNzZXJ0aW9
uICB4bWxuczpzYW1sPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoxLjA6YXNzZXJ0aW9uIiB4
bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiAge
G1sbnM6bGliPSJodHRwOi8vcHJvamVjdGxpYmVydHkub3JnL3NjaGVtYXMvY29yZS8yMDAyLzEy
XJ2bGV0PC9saWI6UHJvdmlkZXJJRD48L2xpYjpBdXRoblJlc3BvbnNlPgo="/>
</FORM>
</BODY></HTML>
```

The corresponding decoded AuthnResponse (line-wrapped) is as follows:

```
<lib:AuthnResponse xmlns:lib="http://projectliberty.org/schemas/core/2002/12"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:samlp="urn:oasis:names:tc
:SAML:1.0:protocol" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="htt
p://www.w3.org/2001/XMLSchema-instance" ResponseID="s823258780a46051de4e643d6a6d
84479db10dbab"  InResponseTo="sa51a95ae420a2a8bb2d608740680c9df6e767dc3"  MajorV
ersion="1"  MinorVersion="0"  IssueInstant="2006-08-01T17:44:16Z"><samlp:Status>
<samlp:StatusCode Value="samlp:Success">
</samlp:StatusCode>
</samlp:Status>
<saml:Assertion  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:xsi="h
ttp://www.w3.org/2001/XMLSchema-instance"  xmlns:lib="http://projectliberty.org/
schemas/core/2002/12"  id="sa3e3d3c81c45413d66bcf6baadeff0624a243e3901" MajorVer
sion="1" MinorVersion="0" AssertionID="sa3e3d3c81c45413d66bcf6baadeff0624a243e39
01" Issuer="https://ide-14.red.iplanet.com:443/amserver/cdcservlet" IssueInstant
="2006-08-01T17:44:15Z" InResponseTo="sa51a95ae420a2a8bb2d608740680c9df6e767dc3"
 xsi:type="lib:AssertionType">
<saml:Conditions  NotBefore="2006-08-01T17:44:15Z" NotOnOrAfter="2006-08-01T17:4
5:15Z" >
<saml:AudienceRestrictionCondition>
<saml:Audience>http://comal-b.central.sun.com:80/?Realm=%2F</saml:Audience>
</saml:AudienceRestrictionCondition>
</saml:Conditions>
<saml:AuthenticationStatement  AuthenticationMethod="LDAP" AuthenticationInstant
="2006-08-01T17:44:15Z" ReauthenticateOnOrAfter="2006-08-01T17:45:15Z" xsi:type=
"lib:AuthenticationStatementType"><saml:Subject   xsi:type="lib:SubjectType"><sa
ml:NameIdentifier NameQualifier="https://ide-14.red.iplanet.com:443/amserver/cdc
servlet">AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7EGE4oJXk%3D%40AAJTSQACMTEAAlMxAA
IwMQ%3D%3D%23</saml:NameIdentifier>
<saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirmatio>
</saml:SubjectConfirmation>
<lib:IDPProvidedNameIdentifier  NameQualifier="https://ide-14.red.iplanet.com:44
3/amserver/cdcservlet" >AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9csUN7EGE4oJXk%3D%40AAJ
TSQACMTEAAlMxAAIwMQ%3D%3D%23</lib:IDPProvidedNameIdentifier>
</saml:Subject><saml:SubjectLocality  IPAddress="192.18.72.87" DNSAddress="ide-1
4.red.iplanet.com" /><lib:AuthnContext><lib:AuthnContextClassRef>http://www.proj
ectliberty.org/schemas/authctx/classes/Password</lib:AuthnContextClassRef><lib:A
uthnContextStatementRef>http://www.projectliberty.org/schemas/authctx/classes/Pa
ssword</lib:AuthnContextStatementRef></lib:AuthnContext></saml:AuthenticationSta>
</saml:Assertion>
<lib:ProviderID>https://ide-14.red.iplanet.com:443/amserver/cdcservlet</lib:Prov>
</lib:AuthnResponse>
```

5. The browser automatically posts the form with LARES to `http://comal-b.central.sun.com:80/agentapp/sunwCDSSORedirectURI` without the user interaction. The agent responds by setting a new SSO token `iPlanetDirectoryPro` with an empty cookie domain. A cookie with an empty DNS domain will have the server FQDN as the domain. Also note the cookie value is exactly the same as the one set in Step 3

response by Access Manager. The only difference is the cookie domain. The HTTP response also redirects the browser to the original requested resource `http://comal-b.central.sun.com:80/app1/test1.html`.

REQUEST:

```
POST /agentapp/sunwCDSSORedirectURI HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Content-Length: 3584
Cookie: amFilterCDSSORequest=AQICAtwmVLBfMe/PgTWWJqWPSfO2eZo6rYLpQLiSI2Uk+Es+I25/
   7Pb5lDpLfNbM1S64amLqY9RLg1i9nEXzWfcnBEVZS5SdG2pJtTdMzEgo/o/
   MARoPq//EMt766UEXFT6aOUAtME0or70=;
   SUN_ID=69.196.39.237:227251153914164
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-Java-System/Application-Server
Date: Tue, 01 Aug 2006 17:44:18 GMT
Content-type: text/html
X-powered-by: Servlet/2.4
Location: http://comal-b.central.sun.com:80/app1/test1.html
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcwzBSR87MxpRCFm
9P5Dx9csUN7EGE4oJXk%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
   Path=/
Set-cookie: amFilterCDSSORequest=reset;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/

Connection: close
```

6. The browser follows the redirection to access the protected resoruce again at `http://comal-b.central.sun.com:80/app1/test.html`. Note the new SSO token is sent to the server. The agent validates the SSO token, evaluates the policies and allows the access. The server responds with the content of the protected resource.

   REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcwzBSR87MxpRCFm9P5Dx9cs
UN7EGE4oJXk%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23;
   SUN_ID=69.196.39.237:227251153914164
```

RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-Java-System/Application-Server
Date: Tue, 01 Aug 2006 17:44:19 GMT
Content-length: 88
Content-type: text/html
X-powered-by: Servlet/2.4
Etag: W/"88-1153320226000"
Last-modified: Wed, 19 Jul 2006 14:43:46 GMT
Connection: close

<html>
<head>
<title>Test1 HTML</title>
</head>
<body>
Test1 HTML
</body>
</html>
```

7. The user now attempts to access `http://am-v210-01.red.iplanet.com:7001/app1/test1.html`. A SSO token is sent with the HTTP request. The browser currently has two SSO Tokens, one for each domain. The token sent was obtained in Step 3. The agent intercepts the request and receives the SSO token. The agent validates the token and permits the server to serve the content of the protected page.

   REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash,
   application/vnd.ms-excel, application/vnd.ms-powerpoint,
   application/msword, */*
Accept-Language: en-us
If-Modified-Since: Tue, 20 Jun 2006 11:03:04 GMT
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-v210-01.red.iplanet.com:7001
Cookie: amservercookie=0C; iPlanetDirectoryPro=AQIC5wM2LY4Sfcwz
BSR87MxpRCFm9P5Dx9csUN7EGE4oJX
   k%3D%40AAJTSQACMTEAAlMxAAIwMQ%3D%3D%23
```

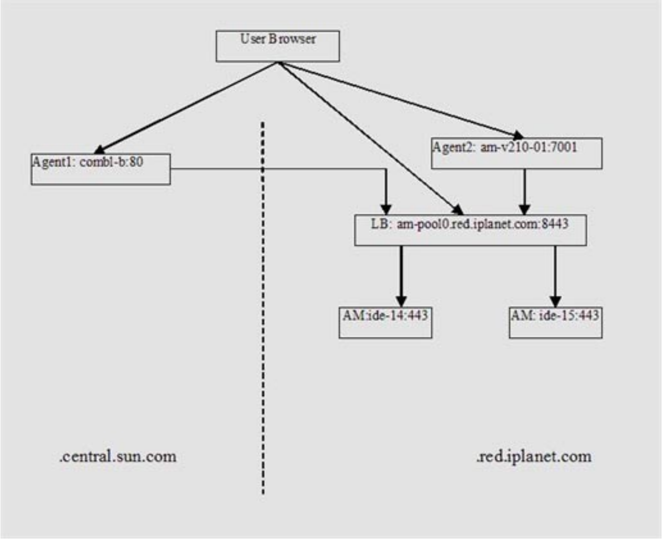
RESPONSE:

```
HTTP/1.1 304 Not Modified
Date: Tue, 01 Aug 2006 17:44:32 GMT
Content-Length: 0
Set-Cookie: JSESSIONID=GPTQKHJWTyvJVSGm31rV59LCzxGTmhqVFfc4GbLY4
L98vBRCYnKT!384704559; path=/
Connection: Close
```

Web Policy Agents Use Case — Protocol Exchange

The following are actual protocol exchanges in two use cases. In both use cases, the configuration are as follows:

- The primary domain (where Access Manager resides) is `.iplanet.com`. The non-primary domain is .sun.com.
- In the primary domain, there are two Access Manager instances are `ide-14.red.iplanet.com:443` and `ide-15.red.iplanet.com:443`. Both are behind a load balancer `am-pool0.red.iplanet.com:8443`.
- In the primary domain, Agent #2 resides in `am-v210-01.red.iplanet.com:7001`with CDSSO disabled because it's in the same DNS domain as the Access Manager instances.
- In the non-primary domain, Agent #1 resides in `comal-b.central.sun.com:80` with CDSSO enabled.
- A protected resource `/app1/test1.html` is deployed on the servers of both agents.



In the use cases, we will demonstrate a CDSSO sequence from the primary domain to the non-primary domain, and the reverse.

Web Policy Agent Use Case 1: Accessing a Protected Resource in the Primary Domain First

1. An unauthenticated user attempts to access `http://am-v210-01.red.iplanet.com:7001/app1/test1.html`. The agent intercepts the request and receives no SSO token. The agent responds with a redirection to the Access Manager login page.

   REQUEST:

   ```
   GET /app1/test1.html HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-v210-01.red.iplanet.com:7001
   ```

   RESPONSE:

   ```
   HTTP/1.1 302 Moved Temporarily
   Date: Thu, 10 Aug 2006 14:44:55 GMT
   ```

```
Location: https://am-pool0.red.iplanet.com:8443/amserver/UI/
    Login?goto=http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001
    %2Fapp1%2Ftest1.html
Content-Type: text/html
Connection: Close

<html><head><title>302 Moved Temporarily</title></head>
<body bgcolor="#FFFFFF">
<p>This document you requested has moved temporarily.</p>
<p>It’s now at <a href="https://am-pool0.red.iplanet.com:8443/amserver/UI/
   Login?goto=http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001%2Fapp1%2Ftest1.html">
   https://am-pool0.red.iplanet.com:8443/amserver/UI/Login?goto=
   http%3A%2F%2Fam-v210-01.red.iplanet.com
   %3A7001%2Fapp1%2Ftest1.html</a>.</p>
</body></html>
```

2. The browser follows the redirection to access the Access Manager login page.

   REQUEST:

```
GET /amserver/UI/Login?goto=http%3A%2F%2Fam-v210-01.red.iplanet.com
%3A7001%2Fapp1%2Ftest1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Host: am-pool0.red.iplanet.com:8443
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Connection: Keep-Alive
```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:44:09 GMT
Content-type: text/html;charset=UTF-8
Cache-control: private
Pragma: no-cache
Expires: 0
X-dsameversion: 7 2005Q4
Am_client_type: genericHTML
Set-cookie: JSESSIONID=D74987DB66D0F603043D1032FF92780D;Path=/;Secure
Set-cookie: AMAuthCookie=AQIC5wM2LY4SfcyUVIxDMmieXosNGE7jBEZdye
Jb0CIYBuc%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23;
   Domain=.iplanet.com;Path=/
Set-cookie: amservercookie=02;Domain=.iplanet.com;Path=/
```

```
<... login page content omitted by authro ...>
```

3. The user types in his credential on the login page and clicks Submit. A login form is posted to Access Manager. If the user authenticates successfully, the Access Manager responds by setting an SSO token (`iPlanetDirectoryPro`) in the domain `.iplanet.com`. The response also redirects the browser to the original requested resource `http://am-v210-01.red.iplanet.com:7001/app1/test1.html`.

   REQUEST:

```
POST /amserver/UI/Login HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Referer: https://am-pool0.red.iplanet.com:8443/amserver/UI/
   Login?goto=http%3A%2F%2Fam-v210-01.red.iplanet.com%3A7001
   %2Fapp1%2Ftest1.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Content-Length: 144
Cache-Control: no-cache
Cookie: JSESSIONID=D74987DB66D0F603043D1032FF92780D;
   AMAuthCookie=AQIC5wM2LY4SfcyUVIxDMmieXosNGE7jBEZdyeJb0CI
   YBuc%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23; amservercookie=02
```

   RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:44:16 GMT
Content-length: 0
Content-type: text/html
Cache-control: private
Pragma: no-cache
X-dsameversion: 7 2005Q4
Am_client_type: genericHTML
Location: http://am-v210-01.red.iplanet.com:7001/app1/test1.html
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcyUVIxDMmieXosNGE7j
   BEZdyeJb0CIYBuc%3D%40AAJTSQACMTE
   AAlMxAAIwMg%3D%3D%23;Domain=.iplanet.com;Path=/
Set-cookie: AMAuthCookie=LOGOUT;Domain=.iplanet.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT;Path=/
Connection: close
```

4. The browser follows the redirection to access `http://am-v210-01.red.iplanet.com:7001/app1/test.html`. Note the SSO token cookie `iPlanetDirectoryPro` is sent in the HTTP request to the server. The agent validates the SSO token and evaluates policies by interacting with the Access Manager in the background. If the access is allowed, the server responds with the content of the protected resource.

   REQUEST:

   ```
   GET /app1/test1.html HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Pragma: no-cache
   Accept-Language: en-us
   Cookie: amservercookie=02; iPlanetDirectoryPro=AQIC5wM2LY4Sfc
      yUVIxDMmieXosNGE7jBEZdyeJb0CIYBuc
      %3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-v210-01.red.iplanet.com:7001
   Cache-Control: no-cache
   ```

   RESPONSE:

   ```
   HTTP/1.1 200 OK
   Date: Thu, 10 Aug 2006 14:45:06 GMT
   Content-Length: 88
   Content-Type: text/html
   Last-Modified: Tue, 20 Jun 2006 11:03:04 GMT
   Accept-Ranges: bytes
   Connection: Close

   <html>
   <head>
   <title>Test1 HTML</title>
   </head>
   <body>
   Test1 HTML
   </body>
   </html>
   ```

5. The user now attempts to access another resource `http://comal-b.central.sun.com:80/app1/test1.html`. Note the SSO token iPlanetDirectoryPro is not sent in the HTTP request because the server `comal-b.central.sun.com` does not match the cookie domain `.iplanet.com`. The agent,

receiving no SSO token, responds by redirecting the browser to the CDC servlet URL `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet`.

REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Cookie: SUN_ID=69.196.39.237:227251153914164
If-Modified-Since: Thu, 10 Aug 2006 14:40:34 GMT
If-None-Match: "23-44db4562"
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:45:15 GMT
Content-length: 0
Content-type: text/html
Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?goto=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html%
   3FsunwMethod%3DGET&RequestID;
   =8382&MajorVersion=1&MinorVersion=0
   &ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Famagent&IssueInstant;
   =2006-08-10T09%3A45%3A16Z
Connection: close
```

The redirection URL contains some parameters to be carried to the CDC servlet. Some of these parameters are:

goto — The URL to which CDC servlet will forward AuthNResponse, which is the original requested URL with a parameter `sunwMethod=GET` appended.

MajorVersion — Major version is set 1. It is Liberty Federation Protocol major version.

MinorVersion — The minor version is set to 1. It is Liberty Federation Protocol minor version.

RequestID — Is the Authn Request ID. It is a randomly generated unique id. This is sent to CDC Servlet so that the its AuthnResponse later can contain this unique identifier. The RequestID is used to tie the response coming back. It is verified when the response comes back from the CDC servlet

ProviderID — It is Service Provider ID - which is the agent. The value will be of the form: `http(s)://<agent-host>:<port>/amagent?Realm=<RealmName>` or `http(s)://<agent-host>:<port>/amagent`, where RealmName is what is configured for property com.sun.identity.agents.config.organization.name in `AMAgent.properties`.

IssueInstant — It is the time at which the AuthnRequest was created, in UTC format.

6. The browser follows the redirection to access the CDC servlet. Note the SSO token iPlanetDirectoryPro is sent in the HTTP request because the server DNS domain matches the cookie domain. The CDC servlet validates the SSO token and responds with a HTML page. The page contains a HTML FORM which will be automatically posted to the agent (`http://comal-b.central.sun.com:80/app1/test1.html?sunwMethod=GET`, based on the "goto" parameter earlier). The form's hidden field LARES is encoded Liberty-like AuthnResponse that contains the existing SSO Token in the domain `.iplanet.com`.

   REQUEST:

   ```
   GET /amserver/cdcservlet?goto=http%3A%2F%
      2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html%3F
      sunwMethod%3DGET&RequestID;=8382&MajorVersion=1&MinorVersion=
      0&ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%
      2Famagent&IssueInstant;=2006-08-10T09%3A45%3A16Z HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
       application/x-shockwave-flash,application/vnd.ms-excel,
       application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   Cookie: JSESSIONID=D74987DB66D0F603043D1032FF92780D; amservercookie=02;
      iPlanetDirectoryPro=AQIC5wM2LY4SfcyUVIxDMmieXosNGE7jBEZdyeJb0CIYBuc%3D%
      40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23
   If-Modified-Since: Thu, 10 Aug 2006 14:40:34 GMT
   If-None-Match: "23-44db4562"
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-pool0.red.iplanet.com:8443
   Connection: Keep-Alive
   ```

   RESPONSE:

   ```
   HTTP/1.1 200 OK
   Server: Sun-ONE-Web-Server/6.1
   Date: Thu, 10 Aug 2006 14:44:27 GMT
   Content-type: text/html
   Pragma: no-cache
   Content-length: 3681
   Connection: keep-alive
   ```

```
<HTML>
<BODY Onload="document.Response.submit()">
<FORM NAME="Response" METHOD="POST" ACTION="http://comal-b.central.sun.com:80/app1
   /test1.html?sunwMethod=GET">
<INPUT TYPE="HIDDEN" NAME="LARES" VALUE="PGxpYjpBdXRoblJlc3BvbnNlIHhtbG5zOmxpYj0ia
HR0cDovL3Byb2plY3RsaWJlcnR5Lm9yZy9zY2hlbWFzL2NvcmUvMjAwMi8xMiIgeG1sbnM6c2FtbD0idXJ
uOm9hc2lzOm5hbWVzOnRjOlNBTUw6MS4wOmFzc2VydGlvbiIgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuY
W1lczp0YzpTQU1MOjEuMDpwcm90b2NvbCIgeG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDk
veG1sZHNpZyMiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY
2UiIFJlc3BvbnNlSUQ9InNmOTgzZjU0NWZlNGQzOWFjMzcyYTZhOWMwNTFhMThiNmZlNjJlMGI0IiAgSW5
...
Nwb25zZVRvPSI4MzgyIiAgTWFqb3JWZXJzaW9uPSIxIiAgTWlub3JWZXJzaW9uPSIwIiAgSXNzdWVJbnN0
YW50PSIyMDA2LTA4LTEwVDE0OjQ0OjI3WiI+PHNhbWxwOlN0YXR1cz4KPHNhbWxwOlN0YXR1c0NvZGUgVm
FsdWU9InNhbWxwOlN1Y2Nlc3MiPgo8L3NhbWxwOlN0YXR1c0NvZGU+Cjwvc2FtbHA6U3RhdHVzPgo8c2Ft
bDpBc3NlcnRpb24gIHhtbG5zOnNhbWw9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjEuMDphc3NlcnRpb2
4iIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiICB4bWxu
czpsaWI9Imh0dHA6Ly9wcm9qZWN0bGliZXJ0eS5vcmcvc2NoZW1hcy9jb3JlLzIwMDIvMTIiICBpZD0icz
cmVkLmlwbGFuZXQuY29tOjQ0My9hbXNlcnZlci9jZGNzZXJ2bGVL2xpYjpBdXRoblJlc3BvbnNlPgo="/>
</FORM>
</BODY></HTML>
```

7. The browser automatically posts the form with LARES to the goto URL `http://comal-b.central.sun.com:80/app1/test1.html?sunwMethod=GET`, without any user interaction. The agent validates the AuthNResponse, and responds by setting a new SSO token iPlanetDirectoryPro with an empty cookie domain. A cookie with no domain will be restricted to the originating server only in the future. Also note the cookie value is exactly the same as the one set in Step 3 response by Access Manager.

   The agent also perform necessary session validation and policy evaluation. If all well, the user is allowed for the access. The protected page is served in the response.

   REQUEST:

```
POST /app1/test1.html?sunwMethod=GET HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Content-Length: 3490
Cookie: SUN_ID=69.196.39.237:227251153914164
```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:45:17 GMT
Content-length: 35
Content-type: text/html
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcyUVIxDMmieXosNGE7j
   BEZdyeJb0CIYBuc%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23;Path=/
Last-modified: Thu, 10 Aug 2006 14:40:34 GMT
Accept-ranges: bytes
Connection: close

Success! This is test1.html page.
```

In responding to this request, the agent goes through the following steps to validate the received AuthnResponse:

a. The status code of the `AuthnResponse` is verified to see if it is successful.

b. The assertions are extracted from the AuthnResponse. There should be only 1.

c. The conditions that are in the assertion are also validated. The main one is the date validity condition. The date validity attributes, not before and notOnorAfter, are verified to verify the assertion has not expired. Hence time synchronization between Access Manager and Agent is crucial.

Web Policy Agent Use Case 2: Accessing a Protected Resource in the Non-Primary Domain First

In this use case, an unauthenticated user first accesses a protected resource in the non-primary domain (`.sun.com`). He then accesses a protected resource in the primary domain (`.iplanet.com`).

1. An unauthenticated user attempts to access `http://comal-b.central.sun.com:80/app1/test1.html`. The agent intercepts the request and receives no SSO token. Because the SSO is enabled, the agent responds with a redirection to the Access Manager CDC servlet URL `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet`.

   REQUEST:

```
GET /app1/test1.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash,
   application/vnd.ms-excel,application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Cookie: SUN_ID=69.196.39.237:227251153914164
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
```

RESPONSE:

```
HTTP/1.1 302 Moved Temporarily
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:47:15 GMT
Content-length: 0
Content-type: text/html
Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?goto=
   http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html%3FsunwMethod%
   3DGET&RequestID;=13293&MajorVersion=1&MinorVersion=0&ProviderID;=http%3A%2F%
   2Fcomal-b.central.sun.com%3A80%2Famagent&IssueInstant;=2006-08-10T09%3A47%3A15Z
Connection: close
```

2. The browser follows the redirection to access the CDC servlet without any SSO token. The CDC servlet responds with a login page.

   REQUEST:

```
GET /amserver/cdcservlet?goto=http%3A%2F%2Fcomal-b.central.sun.com%3A80%
   2Fapp1%2Ftest1.html%3FsunwMethod%3DGET&RequestID;=13293&MajorVersion=
   1&MinorVersion=0&ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%
   2Famagent&IssueInstant;=2006-08-10T09%3A47%3A15Z HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Connection: Keep-Alive
```

   RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:46:27 GMT
Content-type: text/html;charset=UTF-8
Cache-control: private
Pragma: no-cache
Expires: 0
X-dsameversion: 7 2005Q4
Am_client_type: genericHTML
Set-cookie: JSESSIONID=FCD5ED4FC043E1E2C2789D228413DB87;Path=/;Secure
Set-cookie: AMAuthCookie=AQIC5wM2LY4SfcwS5LT8TIP9%2Bs3ZqdIV0aEtBDSLrHxr
%2Fcs%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23
```

```
   ;Domain=.iplanet.com;Path=/
Set-cookie: amservercookie=02;Domain=.iplanet.com;Path=/

<... login page content omitted by the author ...>
```

3. The user types in his credential on the login page and clicks Submit. A login form is posted to Access Manager. If the user authenticates successfully, the Access Manager responds by setting an SSO token (`iPlanetDirectoryPro`) in the domain `.iplanet.com`. The response also redirects the browser back to the CDC servlet `https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet`.

   REQUEST:

   ```
   POST /amserver/UI/Login HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Referer: https://am-pool0.red.iplanet.com:8443/amserver/
      cdcservlet?goto=http%3A%2F%2Fcomal-b.central.sun.com%
      3A80%2Fapp1%2Ftest1.html%3FsunwMethod%3DGET&RequestID;
      =13293&MajorVersion=1&MinorVersion=0&ProviderID;=http%3A%2F%
      2Fcomal-b.central.sun.com%3A80%2Famagent&IssueInstant;=2006-08-10T09%3A47%3A15Z
   Accept-Language: en-us
   Content-Type: application/x-www-form-urlencoded

   Connection: Keep-Alive
   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
   SV1; .NET CLR 1.1.4322)
   Host: am-pool0.red.iplanet.com:8443
   Content-Length: 391
   Cache-Control: no-cache
   Cookie: JSESSIONID=FCD5ED4FC043E1E2C2789D228413DB87;
      AMAuthCookie=AQIC5wM2LY4SfcwS5LT8TIP9%2Bs3ZqdIV0aEtBDSL
      rHxr%2Fcs%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23; amservercookie=02
   ```

   RESPONSE:

   ```
   HTTP/1.1 302 Moved Temporarily
   Server: Sun-ONE-Web-Server/6.1
   Date: Thu, 10 Aug 2006 14:47:53 GMT
   Content-length: 0
   Content-type: text/html
   Cache-control: private
   Pragma: no-cache
   Connection: close
   X-dsameversion: 7 2005Q4
   Am_client_type: genericHTML
   Location: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?
   ```

```
   TARGET=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html
   %3FsunwMethod%3DGET&RequestID;=13293&MajorVersion=1&MinorVersion=
   0&ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Famagent
   &IssueInstant;=2006-08-10T09%3A47%3A15Z
Set-cookie: AMAuthCookie=AQIC5wM2LY4SfcwlpUfPmb1dtNENXWxnAoZSuWvmQ5pg
   UB0%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23;
   Domain=.iplanet.com;Path=/
Set-cookie: amservercookie=02;Domain=.iplanet.com;Path=/
Set-cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcwlpUfPmb1dtNENXWxnAoZSu
   WvmQ5pgUB0%3D%40AAJTSQACMTEAAlMxAAIwMg%;
   Domain=.iplanet.com;Path=/
Set-cookie: AMAuthCookie=LOGOUT;Domain=.iplanet.com;
   Expires=Thu, 01-Jan-1970 00:00:10 GMT;Path=/
```

4. The browser follows the redirection to access the CDC servlet again. This time the SSO token `iPlanetDirectoryPro` is sent in the HTTP request because the server DNS domain matches the cookie domain. The CDC servlet validates the SSO token and responds with a HTML page. The page contains a HTML FORM which will be automatically posted to the URL on the agent (`http://comal-b.central.sun.com:80/app1/test1.html?sunwMethod=GET`, derived from the `goto` and `target` parameters). The form's hidden field LARES is an encoded Liberty-like AuthnResponse that contains the existing SSO Tokein in the domain .iplanet.com.

   REQUEST:

```
GET /amserver/cdcservlet?TARGET=http%3A%2F%
   2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html%3F
   sunwMethod%3DGET&RequestID;=13293&MajorVersion=1&MinorVersion=
   0&ProviderID;=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Famagent&IssueInstant;
   =2006-08-10T09%3A47%3A15Z HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
   application/x-shockwave-flash, application/vnd.ms-excel,
   application/vnd.ms-powerpoint, application/msword, */*
Referer: https://am-pool0.red.iplanet.com:8443/amserver/cdcservlet?
   goto=http%3A%2F%2Fcomal-b.central.sun.com%3A80%2Fapp1%2Ftest1.html%
   3FsunwMethod%3DGET&RequestID;=13293&MajorVersion=1&MinorVersion=0&ProviderID;
   =http%3A%2F%2Fcomal-b.central.sun.com%3A80%2
   Famagent&IssueInstant;=2006-08-10T09%3A47%3A15Z
Accept-Language: en-us
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: am-pool0.red.iplanet.com:8443
Cache-Control: no-cache
Cookie: JSESSIONID=FCD5ED4FC043E1E2C2789D228413DB87;
   amservercookie=02; iPlanetDirectoryPro=AQIC5wM2LY4SfcwlpUfPm
   b1dtNENXWxnAoZSuWvmQ5pgUB0%3D%40AAJTSQACMTEAAlMxAAIwMg%3D%3D%23
```

RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:47:54 GMT
Content-type: text/html
Pragma: no-cache
Content-length: 3685
Connection: keep-alive

<HTML>
<BODY Onload="document.Response.submit()">
<FORM NAME="Response" METHOD="POST" ACTION=
"http://comal-b.central.sun.com:80/app1/test1.html?sunwMethod=GET">
<INPUT TYPE="HIDDEN" NAME="LARES" VALUE="PGxpYjpBdXRoblJlc3BvbnNlIH
htbG5zOmxpYj0iaHR0cDovL3Byb2plY3RsaWJlcnR5Lm9yZy9zY2hlbWFzL2NvcmUvM
jAwMi8xMiIgeG1sbnM6c2FtbD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6MS4wOmFz
c2VydGlvbiIgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjEuMDp
wcm90b2NvbCIgeG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZH
...
NpZyMiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEta
W5zdGFuY2UiIFJlc3BvbnNlSUQ9InM4N2IzNTkzOGRhZjk1YzQ4MTBmYzJlODJkMTFl
MGMyZDI2Y2I4ZDA0IiAgSW5SZXNwb25zZVRvPSIxMzI5MyIgIE1ham9yVmVyc2lvbj0
iMSIgIE1pbm9yVmVyc2lvbj0iMCIgIElzc3VlSW5zdGFudD0iMjAwNi0wOC0xMFQxND
0Nzo1NFoiPjxzYW1scDpTdGF0dXM+CjxzYW1scDpTdGF0dXNDb2RlIFZhbHVlPSJzYW
2FtbDpBc3NlcnRpb24+CjxsaWI6UHJvdmlkZXJJRD5odHRwczovL2lkZS0xNS5yZWQu
Y3NlcnZsZXQ8L2xpYjpQcm92aWRlcklEPjwvbGliOkF1dGhuUmVzcG9uc2U+Cg=="/>
</FORM>
</BODY></HTML>
```

5. The browser automatically posts the form with LARES to the goto URL '`http://comal-b.central.sun.com:80/app1/test1.html?sunwMethod=GET`, without any user interaction. The agent validates the AuthNResponse, and responds by setting a new SSO token iPlanetDirectoryPro with an empty cookie domain. A cookie with no domain will be restricted to be sent to the originating server only in the future. Also note the cookie value is exactly the same as the one set in Step 3 response by Access Manager.

   The policy agent also performs necessary session validation and policy evaluation. If all well, the user is allowed for the access. The protected page is served in the response.

   REQUEST:

   ```
   POST /app1/test1.html?sunwMethod=GET HTTP/1.0
   Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
      application/x-shockwave-flash, application/vnd.ms-excel,
      application/vnd.ms-powerpoint, application/msword, */*
   Accept-Language: en-us
   Content-Type: application/x-www-form-urlencoded
   ```

```
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
SV1; .NET CLR 1.1.4322)
Host: comal-b.central.sun.com
Content-Length: 3482
Cookie: SUN_ID=69.196.39.237:227251153914164

<... posted form omitted by the author ...>
```

RESPONSE:

```
HTTP/1.1 200 OK
Server: Sun-ONE-Web-Server/6.1
Date: Thu, 10 Aug 2006 14:48:44 GMT
Content-length: 35
Content-type: text/html
Set-cookie:iPlanetDirectoryPro=AQIC5wM2LY4SfcwlpUfPmb1dtNENXWx
   nAoZSuWvmQ5pgUB0%3D%40AAJTSQACMTEAAlMxAAIwM=g%3D%3D%23;Path=/
Last-modified: Thu, 10 Aug 2006 14:40:34 GMT
Accept-ranges: bytes
Connection: close

Success! This is test1.html page.
```

Accessing Sun Resources Online

The docs.sun.com (tm) web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-1165-10.