Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Connections 3.0

Abstract

Introduction

This document explains how to set up integration between Lotus Connections 3.0 and Computer Associates SiteMinder. This article looks at this integration almost entirely from the Lotus Connections perspective. For a full policy server, refer to the Lotus Connections 3.0 product documentation topic entitled [Enabling single sign-on for SiteMinder: lc3].

Prerequisites

Before beginning SiteMinder enablement with Lotus Connections 3.0 ensure that the following items are complete:

- Lotus Connections 3.0 is setup and working with the IBM HTTP Server without issue.
- The J2C Authentication Alias "connectionsAdmin" is a user who exists on the LDAP and has administrative rights on the WebSphere Administration Console.

What is SiteMinder

Computer Associates is a Web access control product providing web single sign-on (SSO), centralized policy management for authentication, authorization, auditing and user entitlement.

A SiteMinder Web Agent is a software component that controls access to any resource that can be identified by a URL. The Web Agent resides on a web server and intercepts requests for a resource to determine whether or not the server needs to authenticate and authorize users who request access to the protected web server resources.

When a user requests a page that is protected by SiteMinder, the Web Agent on the HTTP server intercepts the request and prompts the user for authentication. If the user provides valid credentials, the user is authenticated by the WebSphere Application Server. The SiteMinder Trust Association Interceptor (TAI) - also known as Application Server Agent - on the WebSphere Application Server verifies the information in the cookie and sets the User Principal name.

Enterprise Network Deployment with SiteMinder Security Diagram

This configuration diagram shows the Lotus Connections architecture with the addition of Computer Associates SiteMinder to protect the entire configuration. In this scenario SiteMinder is connected to the LDAP and has an agent on the webserver SiteMinder intercepts and presents the SiteMinder login page. Once they authenticate SiteMinder adds its SMSESSION cookie to the request and the user is logged onto Lotus Connections via single-sign on.
How SiteMinder Integration Works

The following diagram explains how SiteMinder Integration works when used with WebSphere TAI and a Web Agent on the IBM HTTP Server.

1. User access protected resource.
   - SiteMinder Web Agent on HTTP Server intercepts the request and prompts for Authentication.
   - User enters Username and password.
2. SiteMinder Web Agent passes username and password to SiteMinder Policy Server.
3. SiteMinder Policy Server attempts to Authenticates the user against the LDAP.
4. Policy Server uses the User Directory Object Details specified in the SiteMinder Administration Console.
5. After successful authentications, the Policy Server Authorizes the user.
   - SiteMinder checks the users and Groups assigned access in the Policy.
   - SiteMinder checks the Rules for the Requested methods and urls.
   - SiteMinder adds SMSESSION cookie to the request.
6. Request is returned to the HTTP Server.
   - SiteMinder Web Agent on the IBM HTTP Server checks for valid SMSESSION cookie.
Request is sent to the WebSphere Server.

1. SiteMinder ASA Agent on the WebSphere Server checks for valid SMSESSION cookie.
2. ASA-Agent asserts user details to the WebSphere Server.
3. WebSphere performs its own internal authorization.
4. Allows access to the requested resource.

1. Response is returned to the Http Server.
2. Response sent to user with the requested resource.

Enabling Single Sign-On with Computer Associate’s SiteMinder

The following section is quite complex, it is therefore recommended to refer to the Lotus Connections 3.0 infocenter along with this guide to get the fullest understanding of how SiteMinder integration with Lotus Connections enablement with Lotus Connections 3.0. For the purposes of this guide the following table represent the values of the various SiteMinder objects required for this configuration.

<table>
<thead>
<tr>
<th>SiteMinder Objects</th>
<th>HTTP Server</th>
<th>WebSphere Application Server Node 1</th>
<th>WebSphere Application Server Node 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Object</td>
<td>connections_wa</td>
<td>node_TAI</td>
<td>node_TAI</td>
</tr>
<tr>
<td>Agent Config Object</td>
<td>connections_wa_conf</td>
<td>node_TAI_conf</td>
<td>node_TAI_conf</td>
</tr>
<tr>
<td>Host Config Object</td>
<td>host_connections</td>
<td>host_node1_TAI</td>
<td>host_node2_TAI</td>
</tr>
<tr>
<td>Trusted Hostname</td>
<td>connections</td>
<td>node1_TAI</td>
<td>node2_TAI</td>
</tr>
</tbody>
</table>

Important Notes

- The connectionsAdmin J2C Authentication Alias that you specified during installation must correspond to a valid account that can authenticate with SiteMinder. It may map to a back-end administrative user account need to update the user ID or credentials for this alias, see the Changing references to administrative credentials topic in the Lotus Connections 3.0 InfoCenter.
- For more information about the SiteMinder Policy Server and Web Agent configuration, go to the SiteMinder BookShelf.
- For more information about the SiteMinder Agent for WebSphere, see the SiteMinder Agent for WebSphere Agent Guide (PDF) and CA SiteMinder Agent for WebSphere Agent Release Notes® (PDF).

You need to create SiteMinder Agent and Domain objects with realms, rules, and a policy that is related to IBM HTTP Server and WebSphere Application Server. When a user requests a page that is protected by SiteMinder, authentication. If the user provides valid credentials, the user is authenticated and an SMSESSION cookie is added to the request which is then passed on to the WebSphere Application Server. The SiteMinder Trust Association Principal that Lotus Connections requires to identify the user.

This task describes a configuration that uses SiteMinder Policy Server 6.0 SP5, SiteMinder ASA 6.0 Agent for WebSphere Application Server (with CR00010 hotfix), and SiteMinder Web Agent v6qmr5-cr035.

To set up SSO using SiteMinder, complete the following steps:

Preparing WebSphere Application Server for SiteMinder

Setup Single Sign-On Domain if not already done so. On the deployment manager navigate to Security -> Global Security -> Web and SIP Security -> Sign Sign-On (SSO). Ensure the following is set:

![Global Security](image)

Next copy the unrestricted JCE policy files to the Application Server and Deployment Manager machines. The unrestricted JCE files can be downloaded from the following web page, note that you will have to login with your /webapp/wlm/web/preLogin.do?source=jcesdk. Once the files are downloaded extract them from the package. The files in question are called :

- US_export_policy.jar
- local_policy.jar

On the two nodes and deployment manager machine go to the following location and take a backup of the existing files and then copy in the new unrestricted versions to this location. All servers, node agents and deployment
Install the Web Agent on IBM HTTP Server

- Download the above version of the Web Agent from the CA website.
- Install the Web Agent. For instructions, go to the SiteMinder BookShelf.
- When you are prompted for the Agent Configuration details, specify the Agent Configuration Object that you created earlier.

Launch the webagent installation wizard, select Next at the below panel:

Accept the license agreement and click next:

Click next at the information panel below:
Select a path to install the webagent to and click next to continue.

Select if you wish a new group to be created in your start menu:

Click Install to begin the Web Agent installation.
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Connect...
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Connection Manager

Input the username and password of the SiteMinder administrator and click next.

Add the IP address of the SiteMinder server:

Input the Trusted Host Name and Host Configuration Object, remember these values are taken from the table at the start of this section and are assumed to already exist before completing this task.

http://www-10.lotus.com/ldd/lcwiki.nsf/dx/Scenario_3_Setting_up_SiteMinder_SSO.html
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...
Input the agent configuration object name:

Agent Configuration Object:

Agent Configuration Object Name: connections_webagent

Select the advanced authentification to use depending on your requirements click Next.

SSL Authentication:

The following SSL configurations are available for this web server. If the Web Agent will be providing advanced authentication, select which configuration you will use to configure IBM HTTP Server 7.0.0:

- HTTP Basic over SSL
- X509 Client Certificate
- X509 Client Certificate and HTTP Basic
- X509 Client Certificate or HTTP Basic
- X509 Client Certificate or Form
- X509 Client Certificate and Form
- No advanced authentification

At the following panel click No, I don't want to configure Self Registration and click next.
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...

http://www-10.lotus.com/ldd/lcwiki.nsf/dx/Scenario_3_Setting_up_Site...
In this case, the installation log at C:\Program Files\netegrity\webagent\install\config_info\CA_SiteMinder_Web_Agent_v6OMRS_InstallLog.log reports the following. There are no fatal errors, so we are safe to proceed:

Installation: Successful with errors.

273 Successes

0 Warnings

11 NonFatalErrors

0 FatalErrors

After configuring the webagent as above, find the WebAgent.conf in the HTTPServer/conf directory. Open this file and edit it so EnableWebAgent=YES. Now restart your HTTP Server. When attempting to access the HTTP Server Splash Screen, this indicates that SiteMinder is set up correctly with the WebAgent.

Install the Application Server Agent

Install the Application Server Agent on your both nodes - node1.example.com and node2.example.com

- Download the above version of the Application Server Agent from the CA website.
- Install the Application Server Agent on each node in your Lotus Connections deployment. For instructions, see the SiteMinder Agent for WebSphere Agent Guide.
- When you are prompted for the Agent Configuration details, specify the Agent Configuration Object that you created earlier.

Launch the installer for the application server agent click next to continue.
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Connect

Visit: http://www-10.lotus.com/ldd/lcwiki.nsf/dx/Scenario_3_Setting_up_SiteMinder_Single_Sign-On_with_Lotus_Connect
Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...

http://www-10.lotus.com/ldd/lcwiki.nsf/dx/Scenario_3_Setting_up_Site...
Allow the wizard time to register the host:

Enter the agent configuration object name and click next:

Review any errors messages in the installation log. In this case there are benign errors. Click Done to exit the wizard.
In this case the installation log at C:\smwasasa\log\CA_eTrust_SiteMinder_Agent_v6.0_for_WebSphere_InstallLog.log reports the following, again there are no fatal errors so we are safe to proceed:

**Summary**

---

**Installation:** Successful with errors.

- **96 Successes**
- **0 Warnings**
- **1 NonFatalErrors**
- **0 FatalErrors**

**Actions on WebSphere Application Server post Agent Installation**

When the Application Server Agent is configured ensure to copy smagent.properties from the agent installation directory - smwasasa\conf to AppServer\profiles\AppSrv01\properties

Configure Trust Association Interceptor on WebSphere Application Server. From the deployment manager administrative console for WebSphere Application Server, click Security > Global security -> Web and SIP security, c
Next Click Interceptors. Click the new button and add an interceptor with the following name (com.netegrity.siteminder.webshpere.auth.SmTrustAssociationInterceptor). Click OK and save the change.

It is OK to delete any unused interceptors - in this case the interceptor we added is the only one required for SiteMinder enablement. Leaving these other interceptors in place will not cause any issues but will result in error messages on saving this change.

Actions on HTTP Server post Agent Installation

Create rewrite rules to remap Atom API requests and to redirect URLs when users log out of Lotus Connections. Open the IBM HTTP Server httpd.conf configuration file. The file is stored in the C:\IBM\HTTPServer\conf directory. httpd.conf below shows these rules implemented in both the HTTP and HTTPS sections of this file. The rules added are shown in bold, your httpd.conf should reflect the below when this step is completed. In this extract the Appendix will be redirected back to the page at home.example.com, which may be a corporate homepage for example. When this change is made save and close the httpd.conf file. Restart the IBM HTTP Server.

<table>
<thead>
<tr>
<th>RewriteEngine on</th>
</tr>
</thead>
<tbody>
<tr>
<td>RewriteCond %{REQUEST_URI} !(.<em>/ibm_security_logout)(.</em>)</td>
</tr>
<tr>
<td>RewriteCond %{QUERY_STRING} !=logoutExitPage=<a href="http://home.example.com">http://home.example.com</a></td>
</tr>
</tbody>
</table>

RewriteCond %{QUERY_STRING} !logoutExitPage=http://home.example.com

Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...
http://www-10.lotus.com/ldd/lcwiki.nsf/dx/Scenario_3_Setting_up_Site...
Configure Lotus Connections Custom Authenticator for SiteMinder

The customAuthenticator element in the LotusConnections-config.xml file defines some key parameters of your single sign-on (SSO) solution. The configuration settings that you can specify in this XML element only affect the customAuthenticator element can differ, depending on the SSO solution that you have implemented. Most attributes are optional, but some might be mandatory in the context of your SSO solution. For more information, see the topic.
Add a SiteMinder authenticator property to the Lotus Connections configuration by editing the LotusConnections-config.xml file. Start the wsadmin client and check out the Lotus Connections configuration file.

Update the custom authenticator values by running the following commands:

Configure the custom authenticator to support server-to-server authentication for SiteMinder:

Set the value of the custom.authenticator.cookieTimeout parameter to be equal to or less than the maximum timeout and idle timeout values already configured. To do this you must open the LotusConnections-config.xml from this file. Specify the timeout value in minutes. In this case 60 minutes is the specified timeout value.

Note: when your production environment is ready, set the AllowSelfSignedCerts property to false. In a similar fashion to adding CookieTimeout this must be done manually on the checked out LotusConnections-config.xml before the reauthentication process.

The below is a snippet of the content of the XML as it should look when updated with the aforementioned values:

```
<customAuthenticator name="SiteMinderAuthenticator">
    <attribute key="AllowSelfSignedCerts" value="true" />
    <attribute key="CookieTimeout" value="60" />
</customAuthenticator>
```

Check the LotusConnections-config.xml file back in by running the following command:

```
wsadmin> XConfigService.checkInConfig("C:/temp","dmCell01")
```

Update the reauthenticate property in the files-config.xml file. When this property is set to false, and when a Lotus Connections application detects a session timeout, users must log in again through the SSO authentication manager.

1. Login to the wsadmin client.
2. Execute the following command to load files administration: `execfile("filesAdmin.py")`
3. Execute the following command to check out the files configuration file: `FilesConfigService.checkOutConfig("C:/temp","dmCell01")`
4. Update the reauthenticate property by running the following command: `FilesConfigService.updateConfig("security.reauthenticateAndSaveSupported", "false")`
5. Check the files-config.xml file back in by running the following command: `FilesConfigService.checkInConfig()`

Enabling and Disabling SiteMinder on the Configuration

If there is need to enable or disable SiteMinder at any point do the following:

1. Edit the WebAgent.conf on the HTTP Server machine (HTTPS//Server/conf/WebAgent.conf) and set "EnableWebAgent=NO". Restart the IBM HTTP Server.
2. Change the custom authenticator back to the default authenticator in the LotusConnections-config.xml.
3. Edit smwasasa/conf/AsaAgent-assertion.conf on both nodes and set "EnableWebAgent=NO".
4. Resynchronise nodes and restart Lotus Connections 3.0.
5. Repeat this process to enable SiteMinder and instead set EnableWebAgent=YES where we set it to NO before. You must also re-enable the custom authenticator in the LotusConnections-config.xml and restart the deployment.

When SiteMinder is enabled the following message should appear in the SystemOut.log for all Lotus Connections application servers to indicate SiteMinder has loaded correctly with the configuration:
Troubleshooting SiteMinder Issues

The following section outlines how to gather trace and log information from the servers for troubleshooting purposes. There is also a common issue outlined with a solution and a brief overview of the key SiteMinder configuration.

Enabling Trace

Most errors encountered in this environment are typically interservice issues - communication errors between the back-end servers often due to authorization issues caused by the introduction of SiteMinder to the configuration. To enable traces, some basic configuration changes are required.

```
com.ibm.connections.httpClient.=all
```

Log Files to Help Diagnose Issues

To get a complete overview of any issues on the system with SiteMinder enabled, the following log files should be consulted:

- **Lotus Connections Server Log Files**
  - SystemOut.log
  - trace.log (if applicable)
- **SiteMinder Log Files (On Nodes...)**
  - smwasasa/log/smasa.log
  - smwasasa/log/sm_tai.log
- **SiteMinder Log Files (On web server...)**
  - netegrity/webagent/log/oa.log
  - netegrity/webagent/log/oa_trace.log
- **SiteMinder Server Log Files**
  - Consult the SiteMinder documentation to uncover what traces and logs can be enabled / referenced on the SiteMinder server side.

Common Issue - HTTP Server does not Start with SiteMinder Enabled

The following error message occurs when attempting to start the HTTP Server.

```
Open the httpd.conf on the webservers and comment out the line containing mod_sm20.dll below:
```

[Application servers > LCContacts_server > Diagnostic trace service > Change log detail levels]

Use log levels to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative: a level near the top of the list includes all the subsequent levels.

![Diagram](image-url)
LoadModule sm_module "C:/Program Files (x86)/netegrity/webagent/bin/mod_sm20.dll"

Now add the below line ending with sm22.dll in its place to load the correct module for SiteMinder Web Agent to work as expected.

LoadModule sm_module "C:/Program Files (x86)/netegrity/webagent/bin/mod_sm22.dll"

SiteMinder Configuration Files created by Web Agent and TAI/ASA

Here is a sample of the key configuration files on the nodes which are correctly configured. Note the relationship between all of the files below. Changes to these files require a restart to the web server in case of web agent:

WebAgent.conf

WebAgent.conf is found in /conf/WebAgent.conf and refers to the AgentConfigObject and SmHost.conf (which contains the policy server connection details). Also note the EnableWebAgent parameter.

# WebAgent.conf - configuration file for SiteMinder Web Agent

# Web Agent Version = 6QMR5, Build = 852, Update = 0

#agentname="<AgentName>, <IPAddress>"

HostConfigFile="/opt/netegrity/webagent/config/SmHost.conf"

AgentConfigObject="connections_wa_conf"

EnableWebAgent="YES"

ServerPath="/opt/IBM/HTTPServer/conf"

localconfigfile="/opt/IBM/HTTPServer7/conf/LocalConfig.conf"

LoadPlugin="/opt/netegrity/webagent/bin/libHttpPlugin.so"

#LoadPlugin="/opt/netegrity/webagent/bin/libAffiliate10Plugin.so"

#LoadPlugin="/opt/netegrity/webagent/bin/libSAMLAffiliatePlugin.so"

#LoadPlugin="/opt/netegrity/webagent/bin/libeTSSOPlugin.so"

#LoadPlugin="/opt/netegrity/webagent/bin/libIntroscopePlugin.so"

Scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...
SmHost.conf
SmHost.conf is found at /bin/SmHost.conf, refers to the policy server by IP address. It also contains the hostname and hostconfigobject reference.

# Host Registration File - SmHost.conf

# This file contains bootstrap information required by

# the SiteMinder Agent API to connect to Policy Servers

# at startup. Be sure the IP addresses and ports below

# identify valid listening Policy Servers. Please do not

# hand edit the encrypted SharedSecret entry.

hostname=node1.example.com

hostconfigobject=host_node_TAI

policyserver=9.162.138.40,44441,44442,44443
scenario 3: Setting up SiteMinder Single Sign-On (SSO) with Lotus Conne...
logfile="YES"

logconsole="NO"

smazconf="/opt/smwasasa/conf/AsaAgent-az.conf"

smauthconf="/opt/smwasasa/conf/AsaAgent-auth.conf"

smassertionconf="/opt/smwasasa/conf/AsaAgent-assertion.conf"

About the Author

Colm O'Brien is a member of the Lotus Connections System Verification Test (SVT) team and works in the area of product deployment and reliability testing. Colm has worked extensively in the area of SiteMinder Single Sign-On.