Cisco BroadWorks

External Portal Integration Guide

Developer's Guide

Release 23.0

Document Version 1

Notification

The BroadSoft BroadWorks has been renamed to Cisco BroadWorks. Beginning in September 2018, you will begin to see the Cisco name and company logo, along with the new product name on the software, documentation, and packaging. During this transition process, you may see both BroadSoft and Cisco brands and former product names. These products meet the same high standards and quality that both BroadSoft and Cisco are known for in the industry.

Copyright Notice

Trademarks

Any product names mentioned in this document may be trademarks or registered trademarks of Cisco or their respective companies and are hereby acknowledged.

Document Revision History

Release	Version	Reason for Change	Date	Author
15.0	1	Created document for Release 14.0 and Release 15.0.	August 27, 2008	Simon Cadieux
15.0	1	Edited changes and published document.	September 25, 2008	Andrea Fitzwilliam
15.0	2	Replaced HTTP_BWS_USERID by HTTPBWUSERID in *Table 1 Custom HTTP Headers* for EV 65118.	December 15, 2008	Goska Auerbach
15.0	2	Edited changes and published document.	January 16, 2009	Andrea Fitzwilliam
16.0	1	Created Release 16.0 from Release 15.0, version 2.	February 7, 2009	Roberta Boyle
17.0	1	Created Release 17.0 from Release 16.0, version 1.	February 2, 2010	Yves Racine
17.0	1	Edited and published document.	March 26, 2010	Margot Hovey-Ritter
17.0	2	Updated section *6.2.6 BWCommunicationUtility/DefaultSettings/External Authentication/EmbeddedAgent* for EV 140594. Replaced references to the Web Server by references to the Xtended Services Platform throughout the document.	April 28, 2011	Goska Auerbach
18.0	1	Updated section *5.2 External Authentication using Web-based Authentication Server* for MR123031-FR118638 – External Authentication Enhancements.	October 3, 2011	Pierre Drapeau
18.0	1	Edited changes and published document.	November 8, 2011	Jessica Boyle
19.0	1	Updated document for Release 19.0.	September 23, 2012	Yves Racine
19.0	1	Edited changes and published document.	October 29, 2012	Patricia Renaud
19.0	2	Updated section *5.2.3.4 Cisco BroadWorks Web Application Session Expiry* for EV 178786.	June 11, 2013	Goska Auerbach
19.0	2	Edited changes and published document.	June 18, 2013	Jessica Boyle
20.0	1	Updated document for Release 20.0, including changes introduced with *MR155587-FR173613-FR175654-FR180801-Single Sign On API For Use With Third Party Web Portals* and *MR168152 - FR173615 – External Authentication For MS Active Directory*.	September 10, 2013	Pierre Drapeau
20.0	1	Edited changes and published document.	October 11, 2013	Joan Renaud
21.0	1	Updated document for Release 21.0.	September 24, 2014	Yves Racine
21.0	1	Edited changes and updated copyright notice.	September 29, 2014	Joan Renaud
21.0	1	Rebranded and published document.	December 9, 2014	Joan Renaud

Release	Version	Reason for Change	Date	Author
21.0	2	Added rebranded server icons and published document.	March 9, 2015	Joan Renaud
22.0	1	Updated document for Release 22.0.	October 7, 2016	Eric Ross
22.0	1	Added section *11 Acronyms and Abbreviations*. Edited changes and published document.	December 12, 2016	Joan Renaud
23.0	1	Updated document for Release 23.0.	September 27, 2018	François Rajotte
23.0	1	Edited changes and published document.	October 11, 2018	Jessica Boyle
23.0	2	Completed rebranding for Cisco and republished document.	March 17, 2019	Patricia Renaud

Table of Contents

Table of Figures

List of Tables

1 Summary of Changes

This section describes the changes to this document for each release and document version.

1.1 Changes for Release 23.0, Document Version 2

This version of the document includes the following change:

- Completed rebranding for Cisco and republished document.

1.2 Changes for Release 23.0, Document Version 1

This version of the document includes the following changes:

- Added External Agent Authentication support for Direct Client Application Integration.
- Removed obsolete application references.

1.3 Changes for Release 22.0, Document Version 1

This version of the document includes the following changes:

- Added details about redundancy for Lightweight Directory Access Protocol (LDAP).
- Corrected minor errors from command line interface (CLI) commands.

1.4 Changes for Release 21.0, Document Version 2

This version of the document includes the following change:

- Added the rebranded server icons.

1.5 Changes for Release 21.0, Document Version 1

There were no changes to this document for Release 21.0.

1.6 Changes for Release 20.0, Document Version 1

This version of the document includes the following changes:

- Added this section *1 Summary of Changes*.
- Updated document for Release 20.0, including changes introduced with *MR155587-FR173613-FR175654-FR180801-Single Sign On API For Use With Third Party Web Portals* and *MR168152 - FR173615 – External Authentication For MS Active Directory*.

2 Purpose

Cisco BroadWorks provides many features that allow for the seamless integration of external portals or third-party applications with Cisco BroadWorks own provisioning interfaces such as the Cisco BroadWorks web portal interface or Open Client Interface (OCI). This allows for the integration of Cisco BroadWorks in an environment comprised of multiple web portals and/or third-party applications while offering the end user an experience that does not require multiple logins (that is, Single Sign-On).

When the external portal or third-party application has knowledge of user passwords, it can leverage normal login transactions and the servlet to integrate to Cisco BroadWorks while passing the known password as a parameter, and eliminate the need for the user to enter it manually.

However, Cisco BroadWorks External Authentication also provides the ability for a customer to store user passwords in a separate database and delegate the authentication of users to an external policy server or authority.

Cisco BroadWorks External Authentication is characterized by:

- The ability to create user accounts in Cisco BroadWorks without passwords, as well as a department administrator, group administrator, and service provider administrator accounts without passwords.
- The ability to integrate an external policy server or authority with Cisco BroadWorks.

Cisco BroadWorks allows for the integration of three types of applications:

- Redirection between web portal applications
- OCI client applications running on end user's system
- Complete third-party systems

This document describes how to use the mechanisms provided by Cisco BroadWorks, including external authentication, in each of the three situations.

3 Overview

Cisco BroadWorks can be integrated to web portal applications, direct client applications, or third-party systems (that is, clients that use the Cisco BroadWorks OCI, including OCI-Client [OCI-C] and/or Open Client Interface-Provisioning [OCI-P]).

The different scenarios can be used simultaneously to build complex applications. The application of the different scenarios is not exclusive.

The end user is either automatically logged in through redirection with ID and password passed as parameters or authenticated on the third-party web portal using the policy server:

- Automatically logged in through redirection with ID and password passed as parameters. In this case, the password is stored on Cisco BroadWorks (that is, external authentication is not used) and the redirecting web portal has knowledge of user passwords. This is shown in *Figure 1.*

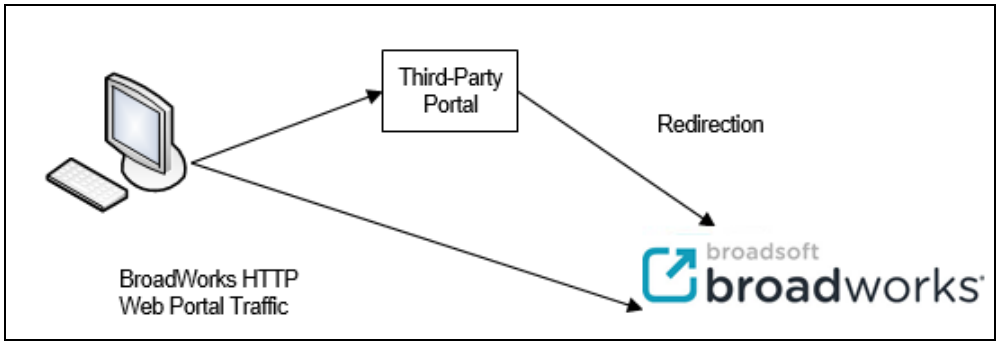

Figure 1 Direct Web Portal Integration through Redirection

- Authenticated on the third-party web portal using the policy server. In this case, the policy server has authority over authentication, and passwords are not stored on Cisco BroadWorks. The same policy server or authority is used when authenticating the user on the Cisco BroadWorks Web Portal upon redirection. This is shown in *Figure 2.*

Figure 2 Web Portal Integration with External Authentication

Figure 3 shows the second situation, in which an OCI-based application is running on the end-user system and connects to Cisco BroadWorks. The password or policy server is used by Cisco BroadWorks to authenticate the end user in the system.

Figure 3 Direct Client Integration

Figure 4 shows a scenario in which an end user connects to a third-party system running outside of Cisco BroadWorks. This third-party system then connects to Cisco BroadWorks through the OCI. In this scenario, Cisco BroadWorks assumes that users have already been authenticated by the third-party system.

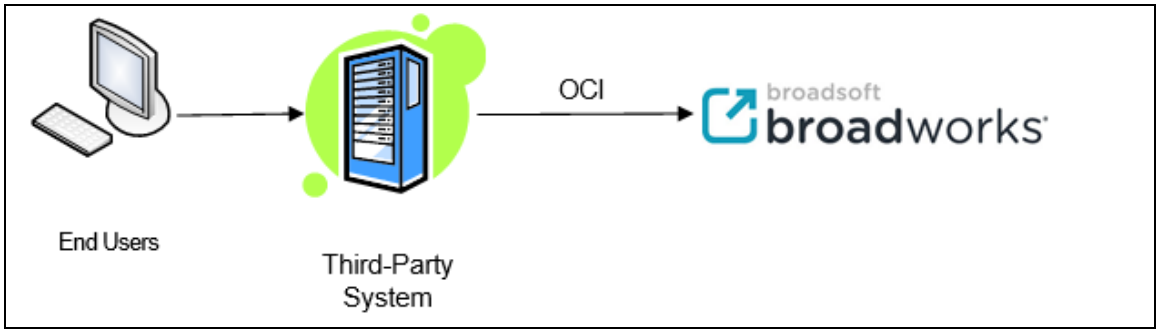

Figure 4 Third-Party System Integration

This section provides a high-level view of the needs for the three types of applications and indicates the various sections that provide more information on the implementation.

3.1 Web Portal Application Integration

Web portal applications are capable of dynamically or statically redirecting a user or administrator to the appropriate Cisco BroadWorks Xtended Services Platform.

3.1.1 Redirection with User ID and Password

If the redirecting web portal has access to passwords stored on Cisco BroadWorks, it can simply redirect the user to a Uniform Resource Locator (URL), giving the user ID, and password as parameters. Therefore, users are not required to enter their passwords twice.

For more information, see section *5.1 Web Portal Integration through Redirection*.

3.1.2 Redirection using External Authentication

Cisco BroadWorks supports two external authentication mechanisms for web portals, the embedded, and the non-embedded mechanisms.

In the *embedded mechanism*, an application (such as the Netegrity SiteMinder suite of products) is packaged on the Cisco BroadWorks Xtended Services Platform, provides protection for web sites, and offers Single Sign-On integration among the Xtended Services Platforms.

In the non-embedded mechanism, the external portal must use a generic authentication mechanism on Cisco BroadWorks, which allows a list of pre-authorized hosts to log in to Cisco BroadWorks without authentication.

Note that regardless of the mechanism used, the Cisco BroadWorks applications, such as Receptionist, can be launched as usual, provided that the proper configuration is applied. For more information, see section *4 External Authentication Prerequisites*.

3.2 Direct Client Application Integration

Cisco BroadWorks allows for external client applications that target end users to connect through the Open Client Server (OCS) or Xtended Services Interface (Xsi) on an Xtended Services Platform (Xsp).

Regular Authentication

Clients can connect to the OCS on Xtended Services Platforms and use regular OCI-P and OCI-C authentication when they have knowledge of user IDs and passwords. Similarly, web applications using Communication Utility use HTTP-based request of some application URL carrying basic authentication credentials when they have knowledge of user IDs and passwords.

OCI authentication is covered in the *Cisco BroadWorks Application Server Provisioning Interface Specification*. HTTP authentication is covered in the *Cisco BroadWorks Computer Telephony Integration Interface Specification*.

Authentication delegated to an external entity

Client applications that do not have access to user passwords can elect to connect to Cisco BroadWorks where an operator has deployed a self-contained and centralized authentication system. In this scenario, it is possible to tie the external password database to the OCS or Xsi for external authentication using any of the following mechanism: Web-based Authentication Server (WAS), LDAP, RADIUS, or Kerberos 5. It is also possible to insert an external agent between client applications and Cisco BroadWorks that takes over all authentication. For the descriptions of these mechanisms, see section *6 Direct Client Application Integration with External Authentication*.

3.3 Third-Party System Integration

3.3.1 Authentication with Single Sign-On

To achieve the specific goal of seamless navigation between a third-party web portal and the Cisco BroadWorks web applications, Single Sign-On (SSO) is implemented, allowing a user to navigate transparently (without being asked for additional credentials) or to be redirected from a web application or portal to a Cisco BroadWorks web application.

This scenario is described in section *7.1 External Authentication using Single Sign On.*

3.3.2 Authentication using Network Access Control List (ACL)

Third-party systems that interact with Cisco BroadWorks through the OCI can be integrated as well. In this scenario, it is assumed that end users are authenticated by the third-party system. The third-party system is considered to be a trusted entity and its address is added to an access control list on Cisco BroadWorks. This scenario is described in section *7.2 External Authentication Using Network Access Control List*.

3.4 Portal Integration Implementation Flowchart

The following flowchart helps to quickly identify the steps required to implement portal integration and refers you to the proper sections in the document.

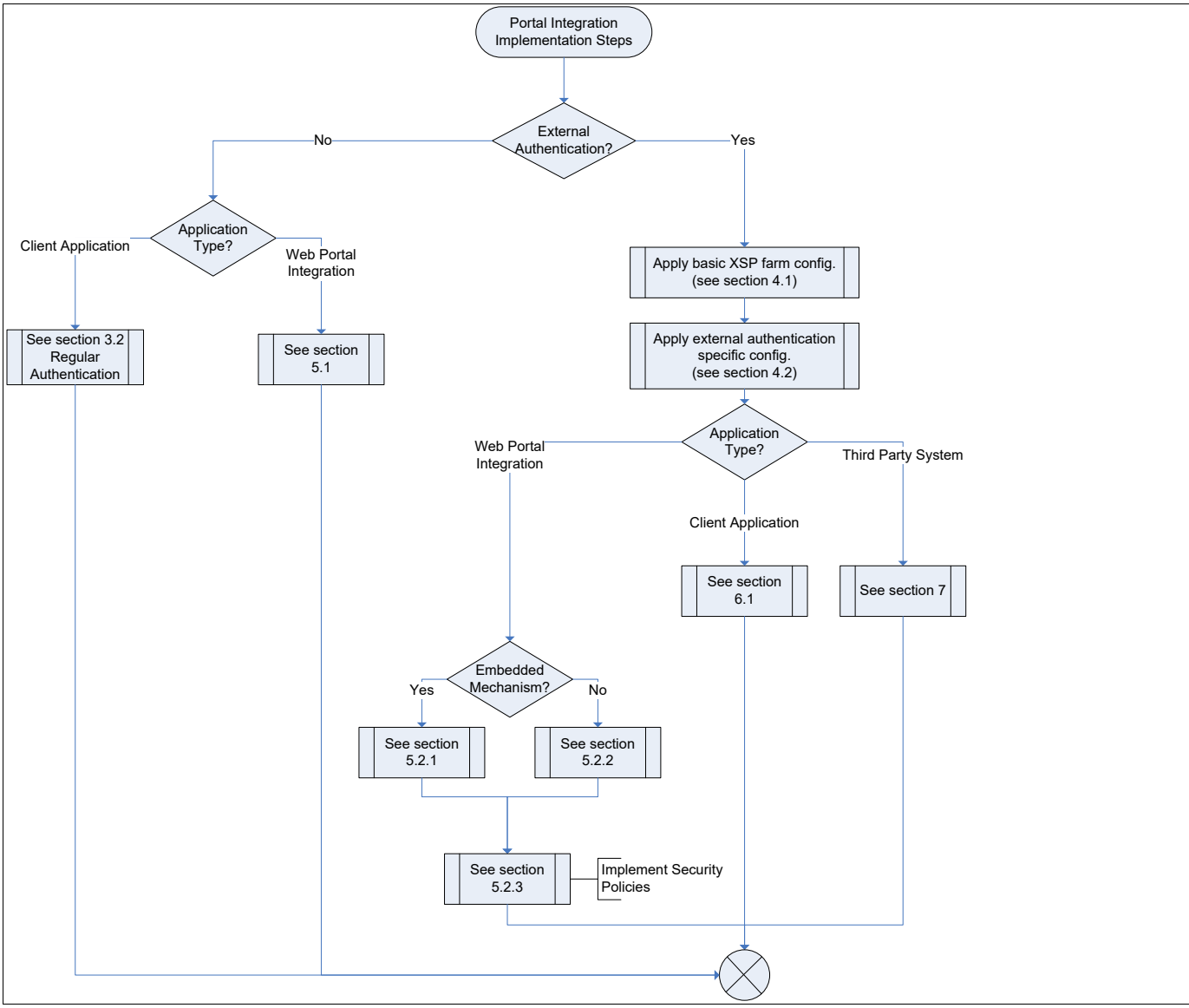

Figure 5 Portal Integration Mechanisms Flowchart

4 External Authentication Prerequisites

4.1 Basic Xtended Services Platform Configuration

All external authentication mechanisms presented in this guide assume that the Xtended Services Platform or the Xtended Services Platform server farm has been properly configured. For information on configuring the Xtended Services Platform, see the *Cisco BroadWorks Xtended Services Platform Configuration Guide*.

4.2 External Authentication Specific Configuration

In addition to the basic Xtended Services Platform configuration, the steps identified in the following subsections must be completed for all external authentication mechanisms to be used.

4.2.1 External Authentication Flag and Password Rules on Application Server

External authentication and password rules can be set at the service provider or system level on every Application Server. Along with the flag that lets customers determine (toggle) whether external authentication is used for a particular set of users, it is also possible to allow administrator and user IDs to be created without an associated password. A user without a password can access Cisco BroadWorks only through external authentication.

External authentication and password rules can be set using the *Administrator* web pages or using the Application Server command line interface (CLI).

Using the Administrator Web Pages

The *Password Rules* web page at the system level, as shown in *Figure 6*, has *Allow users to be created on web portal* check box allowing the administrators to enable or disable the creation of new users without passwords. External authentication can be turned on at the system level when *System and Provisioning Administrators; all other Administrators and Users use external authentication* option is selected.

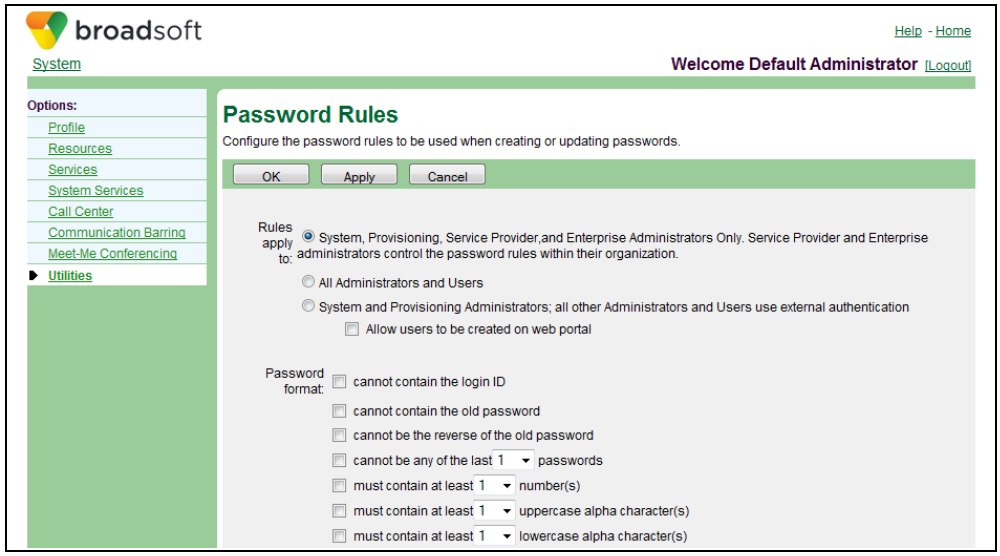

Figure 6 Utilities – Password Rules (System Level)

The *Password Rules* page at the service provider level, shown in *Figure 7*, has two options as well:

- The *Group Administrators and Users use external authentication* button is used to turn external authentication on or off.
- The *Allow users to be created on web portal* check box is used to allow new users to be created without passwords (within this service provider).

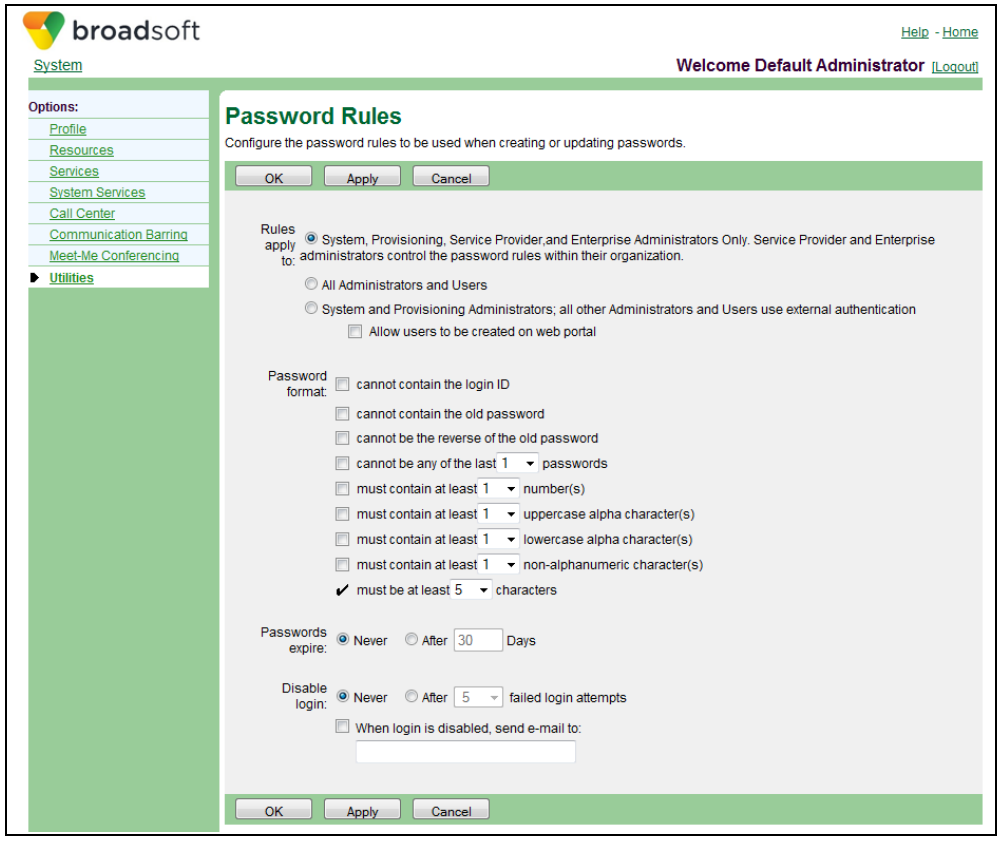

Figure 7 Utilities – Password Rules (Service Provider)

Using the CLI

To set external authentication at the system level, the following Application Server (AS) CLI command must be issued.

```
AS_CLI/SubscriberMgmt/PasswordRules> set rulesApplyTo
sysAdminsOnlyOthersUseExternalAuth
```

To enable the creation of users without passwords (system-wide), the following command must be issued.

```
AS_CLI/SubscriberMgmt/PasswordRules> set allowWebAddExternalUsers true
```

At the service provider level, the following commands are used.

```
AS_CLI/SubscriberMgmt/ServiceProvider/PasswordRules> set
<serviceProvider> rulesApplyTo groupAdminsAndUsersExternalAuth
```

```
AS_CLI/SubscriberMgmt/ServiceProvider/PasswordRules> set
<serviceProvider> allowWebAddExternalUsers true
```

4.2.2 Access Control Lists

First, the loopback address must be added to the external authentication access control list (ACL) on the host where the Open Client Server (and the corresponding Xtended Services Platform) resides. This is because Tomcat on the Xtended Services Platform needs to be able to bypass authentication for OCI-P traffic going through the collocated Open Client Server.

```
XSP_CLI/Applications/OpenClientServer/ExternalAuthentication/AccessContro
lList> add 127.0.0.1 description localhost
```

Secondly, the address of all Xtended Services Platforms and their collocated Open Client Server must be added to the external authentication ACL for **all** target Application Servers. For example, if the Xtended Services Platform server farm is comprised of two servers (192.168.1.1 and 192.168.1.2), then you would enter the following.

```
AS_CLI/System/NetworkAccessLists/ExtAuth> add 192.168.1.1 description
XSP1
Done.

AS_CLI/System/NetworkAccessLists/ExtAuth> add 192.168.1.2 description
XSP2
Done.
```

5 Web Portal Application Integration

5.1 Web Portal Integration through Redirection

Redirecting a user to the Cisco BroadWorks web portal without having to enter the user's password twice is simply a matter of passing the user ID and password as parameters to the Cisco BroadWorks web portal's login servlet. This can be done using Hypertext Transfer Protocol (HTTP) redirection or by securing HTTP if the Cisco BroadWorks web portal has secure HTTP support enabled.

The URL to which the user must be redirected has the following format.

```
http://<xsp_server_farm_fqdn>/servlet/Login?UserID=<userId>&Password=<password>
```

… where:

- `<xsp_server_farm_fqdn>` is the fully qualified domain name (FQDN) of the Cisco BroadWorks Xtended Services Platform server farm.
- `<userId>` is the user ID of the user.
- `<password>` is the password of the user.

5.2 Web Portal Integration using External Authentication

The external authentication mechanism that allows for web portal integration has two variations:

- A specialized (embedded) mechanism based on custom *HTTP* headers.
- A generic (non-embedded) mechanism secured by an access control list.

The specialized mechanism, which is easier to configure and implement, is described first in section *5.2.1 External Authentication using Embedded Agent*, followed by the generic mechanism described in section *5.2.2 External Authentication for Non-embedded Agent*. Section *5.2.3 Security Considerations* addresses the security needed for the solution.

5.2.1 External Authentication using Embedded Agent

This mechanism uses the custom *HTTP* header functionality, provided by applications such as Netegrity SiteMinder Web Agent, to pass the required authentication parameters to the Cisco BroadWorks web application. End users, department administrators, group administrators, and service provider administrators can be authenticated by this mechanism.

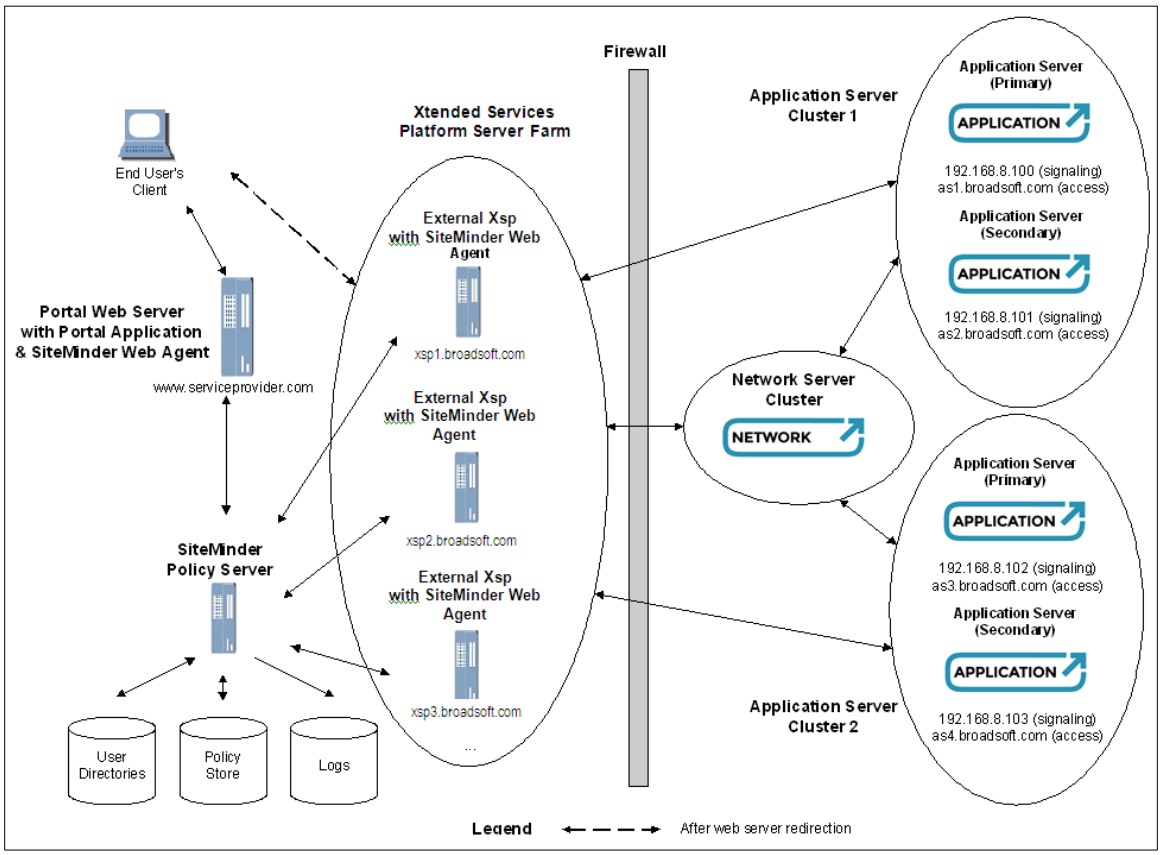

Figure 8 External Authentication Using Embedded Agent

In this scenario, Cisco provides the Application Server cluster, the Network Server cluster, and the Xtended Services Platform server farm. The customer integrates the Cisco BroadWorks web application into the existing portal, developing custom code in the form of a portal application. It is the customer's responsibility to set up an application such as Netegrity SiteMinder that can ensure authentication/authorization and Single Sign-On support for the entire portal, including the Cisco BroadWorks web application.

5.2.1.1 User Login

The steps for external authentication with custom *HTTP* headers for a user (end user or administrator) login are shown in *Figure 9*, which is followed by a description of each step. Note that for simplicity only one Application Server and Xtended Services Platform pair in the Application Server cluster are shown.

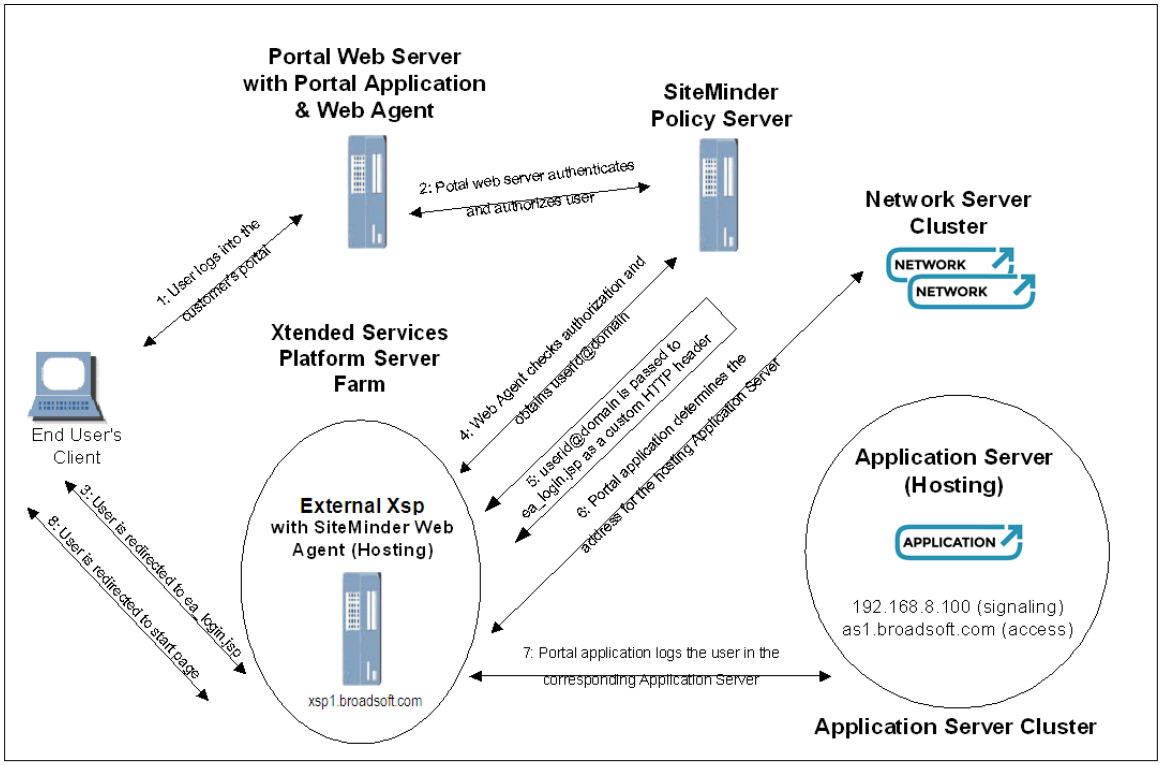

Figure 9 User Login using Custom HTTP Headers

Note that there is a prerequisite for the Cisco BroadWorks system to have external authentication password rules enabled. For more information, see section *4.2.1 External Authentication Flag and Password Rules on Application Server*.

1) The user logs in to the customer's portal web server.

2) The Netegrity SiteMinder application authenticates and authorizes the user. From this point on it is assumed that Single Sign-On is configured among all Xtended Services Platforms involved. At some point, the user clicks on a link for the Cisco BroadWorks CommPilot web application.

3) The portal application redirects the end user's browser to the Xtended Services Platform server farm FQDN or Web Server collocated with an Application Server, and specifically to the Java Server Page (JSP) that expects custom *HTTP* headers, for example:

```
http://xsp_server_farm_fqdn/ea_login.jsp or
http://application_server_cluster_fqdn/ea_login.jsp
```

The Netegrity SiteMinder Web Agent (or equivalent application) running alongside the Xtended Services Platform intercepts the HTTP request and contacts the policy server (or equivalent application) for instructions.

4) The Web Agent checks that the user is authorized to access the Cisco BroadWorks web application.

5) Based on the user's authentication and authorization, the Web Agent inserts custom *HTTP* headers identifying the user to the Cisco BroadWorks web application.

   As part of this step, the Web Agent can optionally specify a "portal login URL" to which the Cisco BroadWorks web application redirects the user when the Cisco BroadWorks web session is terminated. (Usually a user is presented with the *BroadWorks Login* page when the web session is terminated.)

6) The Xtended Services Platform uses the location application programming interface (API) on the Network Server to dynamically resolve the addresses of the Application Servers in the user's hosting cluster.

7) The Xtended Services Platform logs the user in the first available Application Server in the cluster, starting with the primary server.

8) The *ea_login JSP* page redirects the user's browser to the starting page in the Cisco BroadWorks web application, and the user can use all functionality of the Cisco BroadWorks web application.

When there is invalid data in the custom *HTTP* headers (for example, unknown user), the Application Server generates a Simple Network Management Protocol (SNMP) trap, and the user is redirected to the *BroadWorks Login* page. As part of the previous step, the Web Agent can optionally specify an "unknown user URL" where the user is redirected instead.
The same sequence applies whether the user is logging into the Xtended Services Platform or the collocated Web Server on the Application Server. Of course, the Netegrity SiteMinder Web Agent (or equivalent) must be installed alongside the Apache Stronghold web server on the respective machines.

Selected steps are described in more detail in the following subsections.

5.2.1.2 Determine Cisco BroadWorks Application Server Address

The Network Server cluster must have at all times, current information indicating the Application Server cluster hosting a particular user. (In general, the Network Server can contain data from many Application Server clusters.) The Xtended Services Platform uses the Network Server portal interface to determine the Application Server cluster associated with the user that is in the process of logging in.

The Network Server returns the addresses of each individual server in the Application Server cluster hosting a user. The Xtended Services Platform uses the primary server to log the user in, if available, and falls back to the secondary server, if the primary server is not available. In the latter case, the session silently reverts back to the primary server when it becomes available (that is, the user does not notice any disruption in the session). This is true for both administrator and end-user logins.

5.2.1.3 Redirect to ea_login.jsp

The *ea_login.jsp* page is the entry point for the external authentication mechanism using custom *HTTP* headers. The portal application should construct the redirect URL dynamically, based on the address returned by the location API query to the Network Server, as follows.

```
protocol://xtended_services_platform_address/ea_login.jsp
```

The protocol used to build the redirect URL uses the Secure Sockets Layer (SSL) setting on the target Xtended Services Platform. If the SSL setting is "off" or "on", the protocol should be HTTP. If "full" SSL support is enabled on the target Cisco BroadWorks Xtended Services Platform, the protocol should be HTTPS. If HTTPS is not used in the latter scenario, the login still works, but the user is presented with an unnecessary security-related warning in the browser.

5.2.1.4 Insert Custom HTTP Headers

The methodology used to insert the custom *HTTP* headers into the HTTP request depends on the application that is used. The Netegrity SiteMinder Web Agent inserts custom *HTTP* headers when such an instruction is configured in the Response object on the corresponding SiteMinder policy server. The dialog used to insert a dynamic value into a custom *HTTP* header is shown in *Figure 10*.

NOTE: In SiteMinder, the name of the HTTP variable has the "HTTP_" prefix twice.

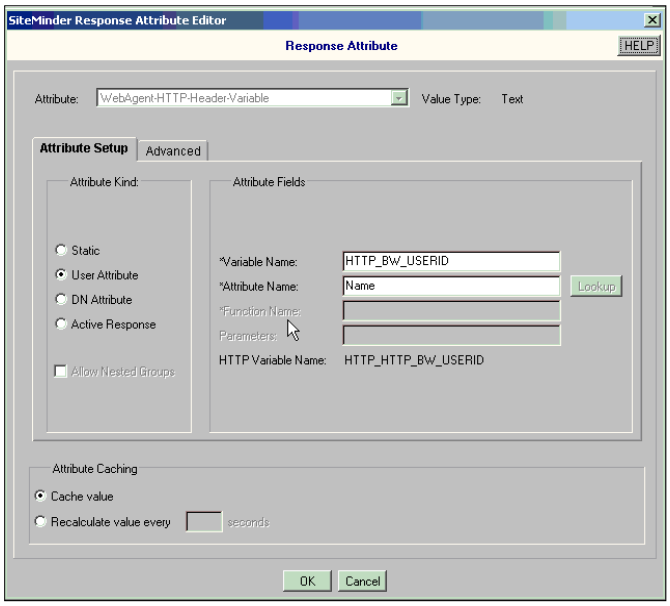

Figure 10 Custom HTTP Header Dialog in Netegrity SiteMinder Policy Server

The custom *HTTP* headers expected by the Cisco BroadWorks web application's ea_login.jsp are listed in *Table 1*.

Name	Default Value	Description
HTTPBWUSERID	None (This header is required.)	The end user's or administrator's user ID in the Cisco BroadWorks system. The value can contain the @domain ending, if required.
HTTP_BW_DOMAIN	""	The part after the "@" sign in the user ID, if required, and if not already provided in the *HTTPBWUSERID* header. Specifically, if a "@" is present in the *HTTPBWUSERID* header value, the HTTP_BW_DOMAIN value is ignored.

Name	Default Value	Description
HTTP_BW_UNKNOWN_USER_URL	By default, error cases are redirected to the Cisco BroadWorks web application login form, at: /Login/index.jsp?EA_ERR=xxx.	The user is redirected to the specified URL when the user ID (and/or domain) are missing or are not recognized by Cisco BroadWorks. An informational alarm is raised. If this URL ends with a "?" character, the appropriate external authentication error code is appended, in the form: *unknownUserURLEA_ERR=xxx.* (For the codes, see section *5.2.1.5 Error Codes*.)
HTTP_BW_PORTAL_LOGIN_URL	/Login/index.jsp or /Logout, depending on the cause of redirection.	When an administrator ends a Cisco BroadWorks session, or when an end user is forced to log in again for some reason, the end user is redirected to this target URL.

Table 1 Custom HTTP Headers

These custom *HTTP* headers are only required for the *ea_login.jsp* page. There is no need to provide these values after that point. Therefore, the policy server can have a specific rule for that page and a general response-less rule for the entire Cisco BroadWorks web application.

5.2.1.5 Error Codes

The following error codes are appended to the *unknownUserURL* (if the URL ends with the "?" character) in case of authentication failure, to help troubleshoot the external authentication problem.

Value	Description
EA_ERR=001	External authentication password rules are not enabled on this system.
EA_ERR=002	Unknown or missing user ID. (This includes the case when the domain is required but not provided, either as part of the user ID or separately.)
EA_ERR=003	The external authentication client is not on the access control list. (This error code does not apply to the custom *HTTP* headers' scenario.)
EA_ERR=004	System error.

Table 2 External Authentication Error Codes

5.2.2 External Authentication for Non-embedded Agent

The generic external authentication mechanism is a JSP that accepts the Cisco BroadWorks userId as an *HTTP* parameter, using a GET or POST method. To prevent unauthorized access, an access control list provides additional protection. The mechanism is HTTP-based; however the external authentication client application does not have to be a browser. The requirements for the external authentication client are to be able to send an HTTP command to the JSP on the Cisco BroadWorks Xtended Services Platform and accept the subsequent HTTP redirection instruction. Such a client could be integrated into the customer's portal application, to be invoked when the user or administrator requests access to the Cisco BroadWorks web application.

5.2.2.1 User Login

The steps for generic external authentication for a user login (end user or administrator) are shown in *Figure 11*. For simplicity, the primary Application Server in the hosting Application Cluster server and one Xtended Services Platform in the Xtended Services Platform server pool are shown.

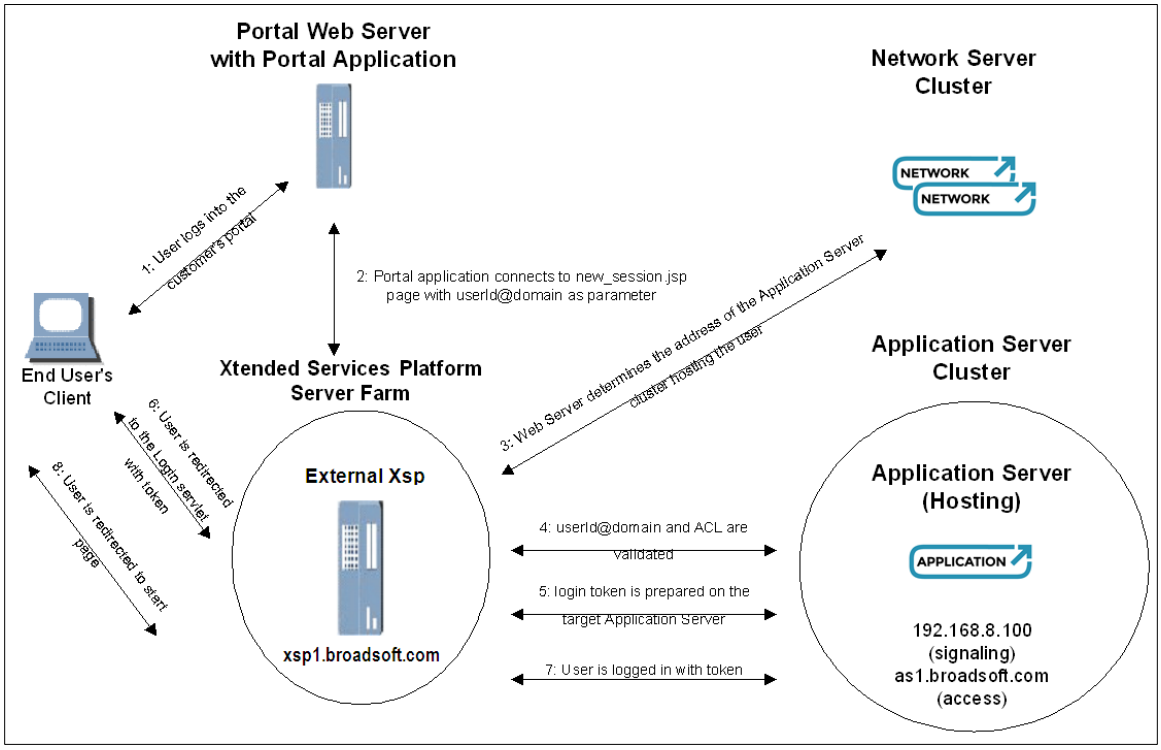

Figure 11 End User Login Using New Session

The prerequisites on the Cisco BroadWorks system are that:

- External authentication password rules are enabled.
- The server on which the portal application is running is added to the external authentication ACL.

1) The user logs into the customer's portal web server. At some point, the user clicks on a link for the Cisco BroadWorks CommPilot web application.

2) The portal application opens a Transmission Control Protocol (TCP) connection to the identified Xtended Services Platform on port 80 (HTTP), and sends the following two-line text request.

```
GET /new_session.jsp?userId=userId HTTP/1.1
Host: address_of_xtended_services_platform
```

NOTE: The parameters can also be sent by the POST method.

As part of this step, the portal application can (optionally) specify a "portal login URL" to which the Cisco BroadWorks web application redirects the user when the Cisco BroadWorks web session is terminated. (Usually, the user is presented with the Cisco BroadWorks *Login* page when the web session is terminated.)

3) The Xtended Services Platform uses the location API on the Network Server to dynamically resolve the addresses of the Application Servers in the user's hosting cluster.

4) The Xtended Services Platform checks that the server hosting the portal application is in the ACL for external authentication, and that external authentication is turned on for the target Application Server.

5) The Xtended Services Platform logs the user in the primary (or secondary, if primary is not available) Application Server of the hosting cluster.

6) The *new_session.jsp* page responds by generating an HTTP redirection response, such as the following.

```
HTTP/1.1 302
Date: date
Server: Stronghold/4.0c …
Set-Cookie: JSESSIONID=…; Path=/
Set-Cookie: JSESSIONID=…; Path=/
Location:
https://address_of_xtended_services_platform/servlet/Login?loginToken
=token&ASAddress=asaddress
Content-Length: 0
Content-Type: text/html
```

When there is invalid data in the custom *HTTP* headers (for example, unknown user), the Application Server generates an SNMP trap, and the user is redirected to the Cisco BroadWorks *Login* page. As part of the previous step, the portal application can (optionally) specify an "unknown user URL" where the user is redirected instead.

7) The portal application parses the response and identifies the redirection URL, which it then presents "as is" to the user's browser. The user either clicks on a link or the user's browser is redirected automatically to that URL by some other mechanism.

8) The Login servlet accepts the *Login* page and redirects the user's browser to the starting page in the Cisco BroadWorks web application, and the user can use all the functionality of the Cisco BroadWorks web application.

The same sequence applies whether the user is logging into the Xtended Services Platform or the collocated Web Server on the Application Server.

Selected steps are described in more detail in the following subsections.

5.2.2.2 Prerequisite for Portal Server in External Authentication ACL

The server on which the portal application is running must be added to the external authentication access control list, using the Application Server CLI from the *AS_CLI/System/NetworkAccessLists/ExtAuth* level.

5.2.2.3 Send HTTP Request to new_session.jsp

The request can be sent over SSL, if SSL support is enabled ("on" or "full") on the target Xtended Services Platform. The supported parameters for the GET or POST method are shown in *Table 3*.

Name	Default Value	Description
userId	None (This parameter is mandatory.)	The end user's or administrator's user ID in the Cisco BroadWorks system. The user ID can contain the @domain, if required.
domain	""	The part after the "@" sign in the user ID, if required, and if not already provided in the *userId* parameter.
unknownUserURL	By default, error cases are redirected to the Cisco BroadWorks web application login form, at: /Login/index.jsp?EA_ERR=xxx	A URL-encoded target URL used when the user ID and/or domain is unknown or missing. If the *unknownUserURL* parameter is provided, it is used instead of the default redirect URL on error. If this URL ends with a "?" character, the appropriate external authentication error code is appended in the form: *unknownUserURLEA_ERR=xxx*. (For the codes, see section *5.2.1.5 Error Codes*.)
portalLoginURL	/Login/index.jsp or /Logout, depending on the cause of redirection.	When an administrator ends a Cisco BroadWorks session, or when an end user is forced to log in again for some reason, they are redirected to this URL-encoded target URL. Note that this value is stored within the user's HTTP session cookie. If the HTTP session expires due to inactivity, the portalLoginURL value cannot be retrieved. In this case, the user is redirected to the default login page of the Cisco BroadWorks portal.
serverName	The server name of the Xtended Services Platform used by the external authentication client (that is, the portal application) to invoke new_session.jsp.	This parameter can be used to force the user to be redirected to the Login servlet on *serverName*, rather than to the Xtended Services Platform through which the portal application connected to *new_session.jsp*. The only meaningful values are the server names of the Xtended Services Platform and the collocated Web Server of the target hosting/primary Application Server.

Table 3 Parameters Supported by new_session.jsp

5.2.2.4 Response from new_session.jsp

The *new_session.jsp* constructs the redirect URL dynamically, based on the following parameters.

```
protocol://address_of_web_server/servlet/Login?loginToken=token&ASAddress
=asaddress
```

The protocol used in building the redirect URL depends on the SSL setting on the target Xtended Services Platform. If the SSL setting is "off" or "on", the protocol is "HTTP". If "full" SSL support is enabled on the target Cisco BroadWorks Xtended Services Platform; the protocol is "HTTPS".

The address_of_web_server is the Xtended Services Platform through which the portal application invoked *new_session.jsp*, unless this is overridden by using the *serverName* parameter.

The token is an internal Cisco BroadWorks value mapped to the userId provided to new_session.jsp. The token allows the user to log in without a password. It is valid for 60 seconds, after which it expires, that is, the portal application must redirect the user to the returned URL within 60 seconds.

The "as address" is the address of the Application Server on which the token has been prepared and is valid.

Note that the value of the *portalLoginURL* parameter can also appear in the redirect URL, under the name "redirectURL", if originally provided when invoking *new_session.jsp*.

5.2.2.5 Redirect User

As mentioned in section *5.2.2.4 Response from new_session.jsp*, the portal application must redirect the user to the returned URL within 60 seconds. If this URL is used after that period, the user is redirected to the *Login* page, unless the *portalLoginURL* parameter was provided to *new_session.jsp*.

5.2.3 Security Considerations

Whichever external authentication mechanism is used, it is important to note the following security concerns:

- SSL support
- Unauthorized login attempts through the Cisco BroadWorks *Login* page
- Cisco BroadWorks password expiry
- Cisco BroadWorks web application session expiry
- Protecting ea_login.jsp from unauthorized access
- Protecting new_session.jsp from unauthorized access
- Cookie issues

This section describes these security concerns.

5.2.3.1 SSL Support

Cisco BroadWorks Application Servers and Xtended Services Platforms support SSL, either on selected pages (SSL is "on") or on all pages in the web application (SSL is "full"). Whether "on" or "full", the SSL setting should be the same in all the servers in a cluster, that is, the two Application Servers and the two Xtended Services Platforms. Otherwise, users receive unnecessary security warnings in their browsers.

5.2.3.2 Unauthorized Login Attempts through Cisco BroadWorks Login Page

The Cisco BroadWorks *Login* page at */Login* can still be accessed when external authentication password rules are enabled. This is because system administrators and provisioning administrators still have to log in through the *Login* page. On the other hand, this means that the *Login* page is exposed to unauthorized login attempts from anyone who has access to it. If required, access to the *Login* page at */Login* should be blocked by some other means (such as Netegrity SiteMinder), for those categories of users that are managed by external authentication.

NOTE: The *Login* page cannot be used if a particular user has a blank password on Cisco BroadWorks. Such attempts are refused, with no additional configuration. The same holds true for the login API part of the Cisco BroadWorks portal API.

5.2.3.3 Cisco BroadWorks Password Expiry

When external authentication password rules are enabled, and if the end user, department administrator, group administrator, or service provider administrator password is blank, then the Cisco BroadWorks password expiry mechanism does not apply. On the other hand, if these users do have a password in Cisco BroadWorks, it can expire based on the password expiry setting, regardless of external authentication.

5.2.3.4 Cisco BroadWorks Web Application Session Expiry

The session timeout for Cisco BroadWorks web application sessions is controlled in the Application Server. The system administrator can decide if sessions expire and configure the timeout. This is configurable at the Application Server CLI, at the *System/ClientSession/InactivityTimer* level.

5.2.3.5 Protect ea_login.jsp from Unauthorized Access

If the custom *HTTP* headers mechanism is not used, but external authentication password rules are enabled, then the *ea_login.jsp* page should be protected from unauthorized access. Using a modified web browser capable of sending the custom *HTTP* headers would enable anyone knowing only the user ID for a user to log in as that end user, department, group, or service provider administrator. Protecting this page can be as simple as adding a redirect statement in the Apache configuration files, sending the requestor to the *Login* page instead.

5.2.3.6 Protect new_session.jsp from Unauthorized Access

Although new_session.jsp is protected by an ACL, this still leaves open the possibility of someone using a web browser from an address in the ACL. If the *new_session.jsp* is invoked from such a browser, any user ID of an existing end user, department, group, or service provider administrator can be provided, and the web browser is redirected to the starting page for that user. Therefore, it is imperative to restrict access to the machine from which *new_session.jsp* is invoked, in terms of using a web browser or installing a browser, if one is not installed.

5.2.3.7 Cookie Issues

For obvious reasons, accessing the Cisco BroadWorks web application through external authentication does not create the user ID/password cookie that is an option when accessing the web application through the *Login* page (that is, by clicking on the *Remember Password* check box). However, if the Cisco BroadWorks system has been upgraded from an existing system on which users or administrators had such cookies and external authentication password rules are enabled, the following restrictions may apply:

- Existing login cookies may not be sufficient to log in.
- Existing login cookies may be invalidated.

5.2.3.7.1 Existing Login Cookies May Not be Sufficient to Log In

When external authentication is enabled in password rules, existing login cookies containing the Cisco BroadWorks user's user ID/password may no longer be sufficient to access the web application, depending on the external authentication configuration and the way the user is accessing the Cisco BroadWorks Login page.

New user ID/password cookies can still be created by logging in through the Login page for the convenience of system administrators and provisioning administrators. However, this is redundant and there is no particular reason to do this for those types of users covered by external authentication. Even then, cookies can only be created by users or administrators whose Cisco BroadWorks passwords are not blank.

5.2.3.7.2 Existing Login Cookies May be Invalidated

In addition to the previous restriction, when external authentication is enabled in password rules, user or administrator passwords can be set to "blank" by the operations support system (OSS) provisioning system. In this case, such user or administrator's existing login cookies are invalidated.

6 Direct Client Application Integration with External Authentication

Cisco BroadWorks OCI-P and OCI-C protocols allowing for third-party clients to perform Cisco BroadWorks provisioning or call control, and web applications such as Call Center, Receptionist, Xsi-Actions, Xsi-Events, and Xsi-MMTel using Communication Utility, do require user authentication.

6.1 Concept Behind External Authentication

Xtended Services Interface (HTTP/CTI interface) and other Web Applications using the Communication Utility, as well as OCS on Xtended Services Platforms support external authentication, which is authentication without involving regular Cisco BroadWorks authentication.

External authentication is achieved by integrating an intermediate server between the password database or policy server and the Xtended Services Interface (Xsi)/Open Client Server.

It is also possible to use an external agent, similar to the mechanism used for Web Portal Application Integration. The external agent is only available on the HTTP interface of the Xtended Services Interface. Enabling the external agent bypasses the authentication for the Xsi and other Web Applications. All incoming requests are trusted. Therefore, additional care must be taken to restrict unauthorized access to the Xsi HTTP interface when this functionality is enabled.

There are four different mechanisms that can be used for external authentication. Each mechanism is discussed in the following sections.

Using any mechanism as an external authentication requires the following configuration parameters to be configured in CommunicationUtility under the following CLI level.

```
/System/CommunicationUtility/DefaultSetting/ExternalAuthentication/> h set

<attribute>, Multiple Choice = {authenticationType, timeout}
  <authenticationType>, Choice = {was, radius, ldap, kerberos5}
  <timeout>, Seconds {5 to 120}
```

Name	Type	Content Restrictions	Default Value	Description
authenticationType	enumeration	WAS, RADIUS, LDAP, KERBEROS5 nillable	nil	This parameter identifies the selected external authentication mechanism. External authentication is disabled when set to nil or left blank.
timeout	Integer	5,120	8	This parameter specifies the timeout (in seconds) for external authentication server queries.

6.1.1 Lightweight Directory Access Protocol (LDAP) for Authentication

The BWCommunicationUtility and Open Client Server act as an LDAP client and communicate with the LDAP server using the URL specified in the Cisco BroadWorks configuration. Based on the server configuration, the LDAP client may need to be authenticated first. Otherwise, the connection is established anonymously.

When LDAP client authentication is required, the Xtended Services Platform uses the security authentication data from the Cisco BroadWorks configuration upon initiating a connection with the LDAP server. The security authentication can have one of the following values:

- Anonymous
- Simple
- Simple Authentication and Security Layer (SASL)

Once the LDAP client is authenticated, the remaining operations relate to authenticating the end user. The connection is closed following each authentication attempt as the LDAP connection is *stateless*.

An LDAP URL must be configured in Cisco BroadWorks and is used to specify how the Xtended Services Platform connects to the LDAP server.

Note that the Cisco BroadWorks configuration captures the following parts:

- ldap scheme (ldap or ldaps)
- hostname
- port

The following is the syntax for the LDAP and Lightweight Directory Access Over SSL (LDAPS) URLs to configure in the Cisco BroadWorks configuration.

```
ldap[s]://hostname:[port]
```

For example:

```
ldap://broadsoft.com:4321
```

6.1.1.1 Redundant LDAP Servers

To achieve redundancy and define multiple LDAP hosts using a single URL, the Xtended Services Platform uses Service Record (SRV) lookups to expand the URL into a list of URLs. SRV lookups are performed when the port is omitted from the configured URL. When using this option, the hostname of the URL is taken as the domain name for a SRV lookup on the following service.

```
_ldap._tcp.<domain>
```

If the port is omitted from the configured URL but no SRV records are found, the URL is used as-is with the default LDAP port (389) or secure port (636), as determined by the scheme of the URL.

The Xtended Services Platform Name Service is used to perform the SRV lookup. As such, the records can either be supplied by an external Domain Name System (DNS) server or by the local *namedefs* file. When a list of SRV records is found by the Xtended Services Platform Name Service, the target of each record is used as the hostname of an LDAP server to use. For each record, a URL is built using the following items:

1) The configured scheme of the URL.
2) The target of the SRV record.
3) The port of the SRV record.

These URLs are used by the Xtended Services Platform to try to connect to an LDAP server. When these resulting URLs are used to connect to an LDAP server, A or AAAA lookups are performed to locate the server. Only the first result of the A/AAAA lookup is used per URL. Therefore, redundancy cannot be achieved by using A/AAAA lookups alone.

The order in which connections are attempted depends on the corresponding priority and weight of the record. The server with the highest priority is always used first. The Xtended Services Platform attempts to connect to the other servers only when a connection fails.

The following example shows a URL that is resolved using SRV lookups.

```
ldaps://broadsoft.com
```

When using this syntax, a lookup on the following SRV record occurs.

```
_ldap._tcp.broadsoft.com SRV 1 10 1234 ldap1.broadsoft.com
_ldap._tcp.broadsoft.com SRV 2 10 636 ldap2.broadsoft.com
```

These records result in the following URLs.

```
ldaps://ldap1.broadsoft.com:1234
ldaps://ldap2.broadsoft.com:636
```

If no SRV records are found, the following URL is used instead.

```
ldaps://broadsoft.com:636
```

6.1.1.2 LDAP Client Authentication

This section summarizes how the Xtended Services Platform (acting as an LDAP client) is authenticated by an LDAP server.

6.1.1.2.1 Anonymous

When the Cisco BroadWorks configuration specifies the security authentication as *anonymous*, the Xtended Services Platform opens a connection to the LDAP server without specifying any credentials. With this client authentication mode, the system uses the *simple* mechanism when subsequently performing end user authentication.

6.1.1.2.2 SASL

When the Cisco BroadWorks configuration specifies the security authentication as "SASL", it must be accompanied by a mechanism name.

- GSSAPI

  The Generic Security Services Application Program Interface (GSSAPI) mechanism is used for Kerberos v5 authentication and optional establishment of a security layer. The Xtended Services Platform creates an authorization ID based on the principal and credentials provided in the Cisco BroadWorks configuration. The LDAP server validates the authenticity by comparing the authentication ID generated locally with the authentication ID received from the client.

  When using GSSAPI, the URL must be configured with an FQDN and an Internet Protocol (IP) address is not permitted.

  Note that the value of the principal corresponds to a bare user name. Authentication with GSSAPI does **not** use the format of DNs.

- DIGEST-MD5

  The DIGEST-MD5 mechanism allows the encryption of the data exchanged between the client and the LDAP server even when using an unsecure link. The Xtended Services Platform only exchanges encrypted data with the LDAP server. The approach for client authentication is similar to LDAP simple, where an administrator has to configure a principal and a credential to the Cisco BroadWorks configuration. The value of the principal is expected to specify an LDAP user with administrative rights. The value of the credentials is expected to hold the password associated with the principal. The LDAP server validates the authenticity by comparing the authentication ID generated locally with the authentication ID received from the client.

  Note that the value of the principal corresponds to a bare user name. Authentication with DIGEST-MD5 does **not** use the format of DNs. Furthermore, in the context of the Microsoft (MS) Active Directory, the principal must not be any user with an attribute marked as *isCritialSystemObject=TRUE*. For example, the principal cannot be the default administrator created by MS Active Directory.

6.1.1.3 End-User Authentication

Once the LDAP server authenticates the Xtended Services Platform as an LDAP client, user authentication may occur.

The end user authentication step concerns mapping authentication identities to LDAP DNs, and depends on how entries are laid out in the LDAP directory.

When making a connection to search the LDAP directory and retrieve user and role information, the Xtended Services Platform extracts information from the Cisco BroadWorks configuration where the LDAP URL is defined to specify the following aspects:

- The domain name of the directory server to which to connect.
- The port number (optionally).

When the default MS Active Directory configuration is used, a login user may be authenticated directly to a *userPrincipalName* in the MS Active Directory.

When the default MS Active Directory is not used, the following additional parameter needs to be configured:

- The Distinguished Name (DN) mapping of the required root naming context.

Doing so enables user search in LDAP which provides greater flexibility, however it also directly impacts the overall performance of queries as the time incurred to perform an LDAP query may increase significantly. This performance is impacted further by enabling the subtree search. Note also that the security level is ultimately dictated by the LDAP configuration choices made by Cisco customers.

The following subsections describe two possible mappings. The first one is associated with the non-SASL security authentications. The second mapping concerns the SASL security authentications. The configuration data for *userToDnMapping* (*BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping*) provides a flexible mechanism for providing such mapping.

6.1.1.3.1 User to DN Mapping for Non-SASL Security Authentication

This DN root naming requires specifying how a login user maps to an LDAP DN.

Often the DN of the user's entry contains the user name presented for authentication but is otherwise the same for all users. In this case, the Cisco BroadWorks configuration provides the *userPattern* data that can be used to specify the DN, with "{0}" marking where the user name should be substituted.

Otherwise, the Xtended Services Platform must search the LDAP directory to find a unique entry that contains the user name. The Xtended Services Platform provides the following configuration data to define the user search:

- *userBase*: This is the entry that is the base of the subtree that contains users. If not specified, the search base is the top-level context.
- *userSearch*: This pattern specifies the LDAP search filter to use after substitution of the user name.
- *userSubtree*: This is the search scope. When set to "true", it searches the entire subtree rooted at the *userBase* entry. The default value of "false" requests a single-level search including only the top level.

Once the Xtended Services Platform obtains a unique match, the Xtended Services Platform must determine how to build the corresponding DN. The system provides additional flexibility in case a non-standard DN structure is used. In such a case, the *userPattern* field can be used to associate a specific attribute with the attribute value received from the search. That is, the keyword "{0}" identifies the user name location and must be preceded by the equal sign "=" and the attribute name. The rest of the *userPattern* provides the rest of the DN. For example, after setting the *userPattern* to "uid={0},ou=people,dc=broadsoft,dc=com", the system fetches the value associated with the *uid* attribute, substitutes the result in the *userPattern*, and then sends it for authentication.

When a standard DN structure is used, there is no need to configure the *userPattern*. In these cases, the Xtended Services Platform uses the default DN received with the unique match and sends it for authentication.

6.1.1.3.2 User to DN Mapping for SASL Security Authentication

For SASL security authentication, the mapping specifies how a login user maps to a SASL user name.

If the login user maps directly to the SASL user name, no configuration is required.

Otherwise, the Xtended Services Platform must search the LDAP directory to find a unique entry containing the user name. The Xtended Services Platform provides the following configuration data to define the user search.

- The *userBase*: This is the entry that is the base of the subtree that contains users. If not specified, the search base is the top-level context. Once the Xtended Services Platform obtains a unique match, it binds to the LDAP server by concatenating the "uid" common name to the user name value (that is, uid=<username>) and prefixing it to the *userBase* (that is, uid=<username><userBase>).
- The *userSearch*: This pattern specifies the LDAP search filter to use after substitution of the user name.
- The *userSubtree*: This is the search scope. When set to "true", it searches the entire subtree rooted at the *userBase* entry. The default value of "false" requests a single-level search including only the top level.

Once the search completes, the Xtended Services Platform uses the *userPattern* data to identify the specific attribute from the DN to use for authentication. That is, the keyword "{0}" identifies the user name location and must be preceded by the equal sign "=" and the attribute name. If the *userPattern* is not configured, the Xtended Services Platform uses the value associated with the *uid* attribute or the *cn* attribute (if the *uid* attribute is not available). For example, after setting the *userPattern* to "cn={0}", the system fetches the value associated with the *cn* attribute and sends it for authentication.

If an LDAP search yields multiple results, a log is generated to warn the administrator about a possible mismatch in the search result. However, the processing always uses only the first result.

6.1.1.4 LDAP Data Interchange Format

The LDAP Data Interchange Format (LDIF) is a standard plain text data interchange format for representing LDAP directory content.

Each content record is represented as a group of attributes, with records separated from one another by blank lines. The individual attributes of a record are represented as single logical lines (represented as one or more multiple physical lines via a line-folding mechanism), comprising "name: value" pairs. Value data that does not fit within a portable subset of ASCII characters are marked with "::" after the attribute name and encoded into ASCII using base64 encoding. The content record format is a subset of the Internet directory information type.

The following list describes the *LDIF* fields.

- dn: distinguished name

  This refers to the name that uniquely identifies an entry in the directory.

- dc: domain component

  This refers to each component of the domain. For example, www.broadsoft.com is written as DC=www,DC=broadsoft,DC=com.

- ou: organizational unit

  This refers to the organizational unit (or sometimes the user group) to which the user belongs. If the user belongs to more than one group, you may specify as such, for example, OU= Engineering,OU=Platform.

- cn: common name

  This refers to the individual object (person's name, meeting room, recipe name, job title, and so on) for whom/which you are querying.

6.1.1.5 Example: External Authentication with LDAP

For the complete list of LDAP setup examples and their relevant Cisco BroadWorks configuration, see *Appendix B: External Authentication with LDAP Examples*.

6.1.2 Remote Authentication Dial-In User Service (RADIUS)

Within Cisco BroadWorks, RADIUS is a networking protocol used to authenticate users or devices before granting them access to the network.

The BWCommunicationUtility and Open Client Server act as Remote Access Servers (RASs) – the RADIUS client – and communicate with the RADIUS server using the host name and port specified in the Cisco BroadWorks configuration. Note that from a client perspective, the RAS does not require specific authentication. However, the RADIUS server is expected to validate the RAS legitimacy by using an access list.

6.1.2.1 Authentication

From an end-user perspective, the same behavior still applies. That is, a user tries to authenticate, either through a browser-based (HTTP/HTTPS) connection (for Xtended Services Interface) or through a connection request to the Cisco BroadWorks Open Client Server.

Upon receiving a login request from an end user, Cisco BroadWorks creates an encrypted *accept-request* message using the following data:

- User credentials
- RADIUS authentication scheme (Password Authentication Protocol [PAP], Challenge-Handshake Authentication Protocol [CHAP], or Microsoft Challenge Handshake Authentication Protocol Version 2 [MS-CHAP v2])
- RADIUS-shared secret

Upon receiving the RADIUS *accept-request*, the RADIUS server performs the following steps:

1) It validates the RADIUS client legitimacy: Is the Xtended Service Platform a known client to the RADIUS server?
2) It validates that the shared secret is correct.
3) It determines whether the provided authentication scheme is supported.
4) It validates the user credentials.

If the RADIUS *accept-request* meets all the above conditions, the RADIUS server sends an *access-accept* message back to the Xtended Service Platform.

If the RADIUS *accept-request* fails any of the above conditions, the RADIUS server sends an *access-reject* message that shows authentication failure.

1) When the Open Client Server receives an *access-reject* message that includes a reply message, the Open Client Server forwards it back to the requestor. (The term *requestor* refers to any of the software clients that interact with RADIUS.)
2) When the CommunicationUtility receives an *access-reject* message that includes a reply message, it is ignored.

The following figure shows the list of possible messages exchanged between the RADIUS client (RAS) and the RADIUS server.

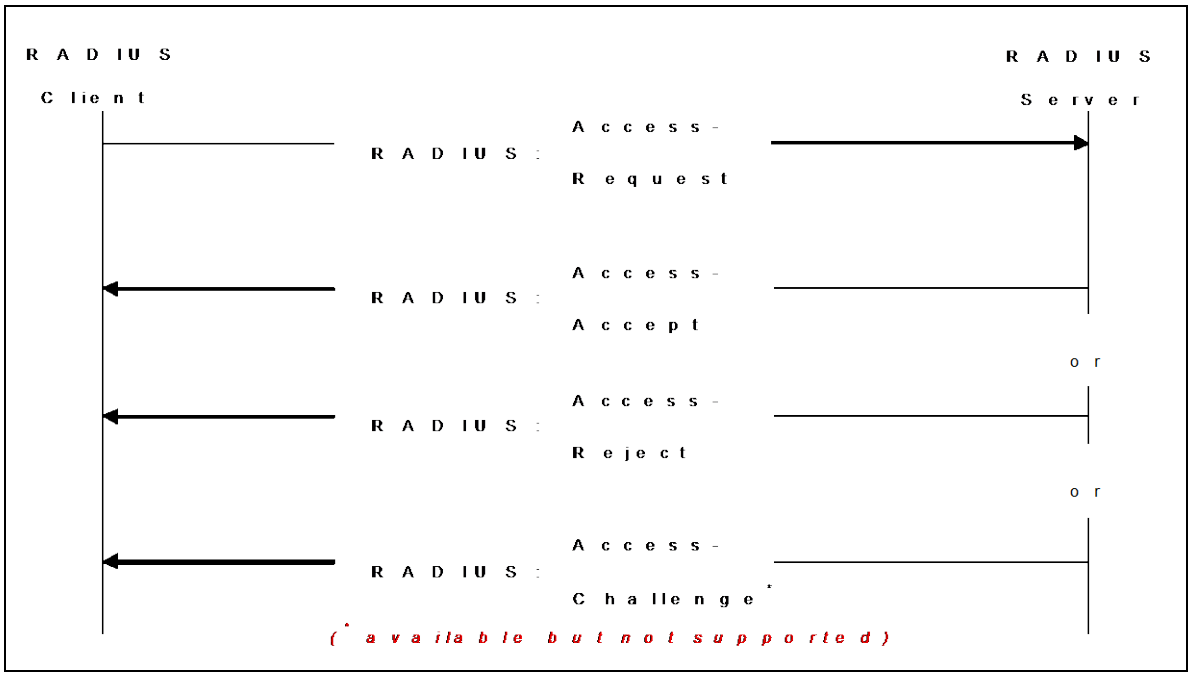

Figure 12 RADIUS Client-Server Authentication Message Flow (Adapted from RADIUS)

6.1.2.2 Authentication Scheme

The following authentication schemes are supported:

- PAP
- CHAP
- MSCHAPv2 (this is the MS Active Directory support for the RADIUS CHAP authentication scheme)

6.1.3 Kerberos 5 as a Stand-Alone External Authentication Service

Kerberos 5 is a security service available as a stand-alone external authentication service.

The *BWCommunicationUtility* and Open Client Server act as a Kerberos 5 client and communicate with the Kerberos 5 server. A key distribution center (KDC) identifies the location of the Kerberos 5 server. An administrator must specify the KDC host name and port in addition to a Kerberos realm in the Cisco BroadWorks configuration. From a Cisco BroadWorks perspective, the Kerberos 5 client does not need to be authenticated first as the connection is used purely to authenticate end users.

Hence, the Xtended Services Platform does not require security authentication data from the Cisco BroadWorks configuration.

6.1.3.1 Kerberos Configuration File

The support of a Kerberos 5 client requires the use of a configuration file. The file must be available to the Cisco BroadWorks BWCommunicationUtility. To provide support for Kerberos 5 from a non-Cisco BroadWorks server, the file location is made available from the BWCommunicationUtility properties file using the property named *bwIntegration.kerberos5.config.file.location*.

6.1.4 Web-based Authentication Server (WAS)

The Xtended Services Interface and Open Client Server can be set up to interact with an external authentication authority using a proprietary protocol built on HTTP requests and responses (Web-based Authentication Server). In this case, the Xtended Services Interface and Open Client Server relies on the WAS to provide authentication services. In this scenario, the customer provides the WAS, implementing the protocol described in the following figure.

This mechanism involves the use of a Web-based Authentication Server, which serves as an external authority to authenticate users.

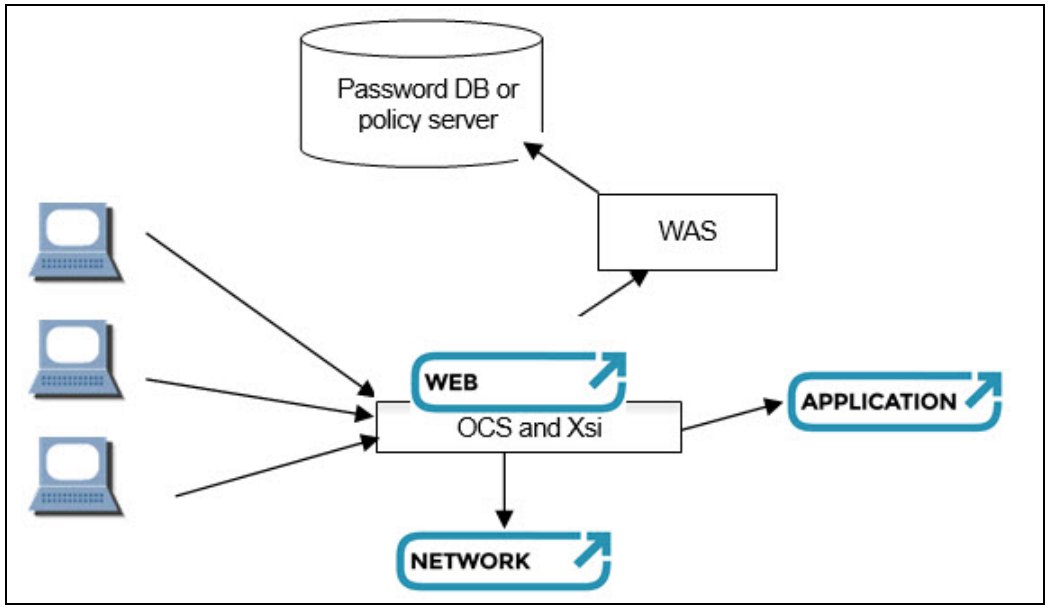

Figure 13 OCI Client External Authentication through WAS

The role of the WAS is to act as an intermediate server between the Xtended Services Interface and Open Client Server, and an External Authentication policy server/service. The WAS is used to abstract or provide a front end to any authentication system (that is not provided by Cisco) used by a customer. The customer is responsible for providing a WAS-compliant application that interacts with their own policy server.

The interaction between client applications and the Xtended Services Interface/Open Client Server is unchanged **even** when the external authentication mode is enabled, that is, there is no impact on either the externally exposed OCI-P or the OCI-C API or the Xsi HTTP GET request of some URL application.

6.1.5 Communication Protocol for the Web Authentication Server

The communication protocol for doing authentication with a WAS is identical for OCS and Xtended Services Interface, and is done over HTTP, as a sequence of two GET requests (header without a body) and responses (header and body), the response body containing the result of the authentication, in XML format:

1) Authenticate (request)

   Example:

```
http://server/servlet/authenticationservlet
?operation=authenticate
&userId=user@domain
```

2) Authenticate (response)

Example:

```
200 OK
<?xml version="1.0" encoding="UTF-8" ?>
<com.broadsoft.protocols.extauth.AuthenticationResponse
errorCode="0"
nonce="12345678" />
```

3) Login (request) – Optional *BasicAuth Credentials*

Example:

```
http://server/servlet/authenticationservlet
?operation=login
&userId=user@domain
&hashedPwd= f7d483bd91499a340b0dea814d05686c
&basicCredentials=YVVzZXIxQGxvY2FsaG9zdDphdXNlcjE=
```

4) Login (response)

Example:

```
200 OK
<?xml version="1.0" encoding="UTF-8"?>
<com.broadsoft.protocols.extauth.LoginResponse errorCode="0"/>
```

The parameter *basicCredentials* can be provided to the WAS only if the configuration parameter *supportsBasicCredentials* is set to "true". When *supportsBasicCredentials* is set to "true", the parameter *basicCredentials* will be included in the request to the WAS always if using the CommunicationUtility (Xtended Services Interface and other web applications), and it will only be included for OCS-based clients if those include the *clearTextPassword* in their request to the OCS.

The parameter *hashedPwd* is always provided in the login request. Because *hashedPwd* is always provided, either the OCS client application or the Communication Utility (on behalf of the web application) needs a nonce to build it, regardless of whether this parameter is later used by the WAS to perform authentication.

Therefore, the WAS must provide a nonce in its authenticate response to respect the protocol requirements (provided that the user is known by the WAS). This nonce is considered "dummy" if the WAS is designed to respond to a login request using basic credentials (plainTextPassword), and therefore, and only in such a case, it could be set to any fixed value.

The following is a list of two definitions:

- OCI-P Signed Password

  Signed Password = MD5 (nonce + ":" + SHA (password))

- Basic Authentication Credentials

  BasicAuth Credentials = Base64 (userId + ":" + password)

6.1.6 Cisco BroadWorks Xtended Services Interface/Communication Utility: Support for HTTP When Using a Web Authentication Server

Cisco BroadWorks Communication Utility is able to perform HTTP authentication using a WAS and the following describes the mechanism behind it.

From the client's perspective (Xtended Services Interface application), the authentication sequence consists of one single step.

An Xtended Services Interface application request will typically be an HTTP GET of some URL of the application, carrying basic authentication credentials.

If WAS (External Authentication) is enabled, the Xtended Services Interface tries to authenticate the user with the WAS. The Xtended Services Interface communicates with the WAS using the protocol described in the previous section.

Xtended Services Interface builds a Hashed Password (*hashedPwd)* and, provided the config parameter *supportsBasicCredentials* is set, *BasicAuth Credentials* and forwards it (them) in an HTTP GET login request to the WAS. If the WAS (External Authentication) is not configured or the user is not known by the WAS, the authentication falls back to default Cisco BroadWorks Network Server/Application Server authentication.

Once the authentication is successful, the Xtended Services Interface proceeds with the HTTP GET of the above-mentioned application URL.

The figures in the following sections illustrate the three possible scenarios in Xtended Services Interface authentication using a WAS UserId known by WAS – authorization success, UserId known by WAS – 401 authentication challenge, and a UserId unknown by WAS.

6.1.6.1 Xsi Application to Xsi Using External Authentication: UserId Known by WAS – Authorization Success

The following figure describes the successful proceeding of an HTTP GET request carrying the credentials of a user successfully authenticated by the WAS.

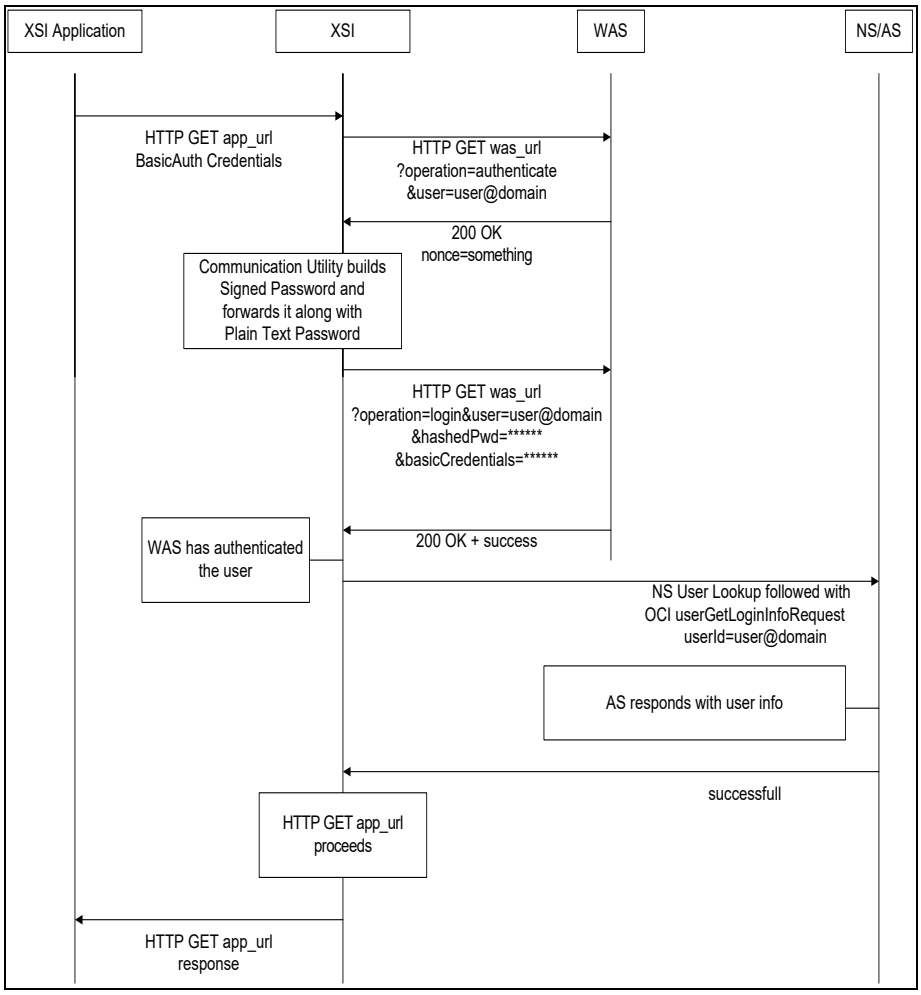

Figure 14 Xsi Using External Authentication – WAS Success Authentication

6.1.6.2 Xsi Application to Xsi using External Authentication: UserId Known from WAS – 401 Authentication Challenge

The following figure describes the authentication challenge of an HTTP GET request carrying the credentials of a user known by the WAS but unsuccessfully authenticated, either because the credentials are incorrect (Case 1) or the user is unknown by Cisco BroadWorks, even though the WAS was able to authenticate the user (Case 2).

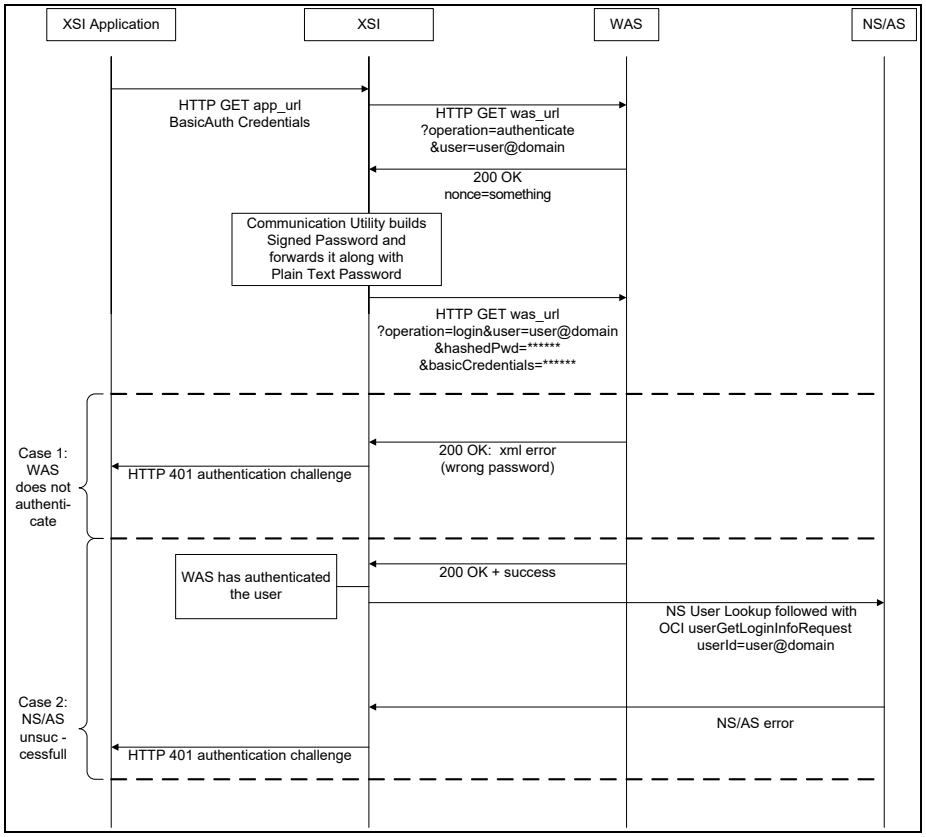

Figure 15 Xsi Using External Authentication – User Known by WAS With Unsuccessful Authentication

6.1.6.3 Xtended Services Interface Application to Xtended Services Interface Using External Authentication: UserId unknown from WAS

The following figure describes the proceeding of an HTTP GET request carrying the credentials of a user unknown by the WAS with a fallback to a successful Network Server/Application Server user authentication (Case 1). The figure also describes an authentication challenge of an HTTP GET request carrying the credentials of a user unknown by the WAS with a fallback to an unsuccessful Network Server/Application user authentication, either because the credentials are incorrect or the user is unknown by Cisco BroadWorks (Case 2).

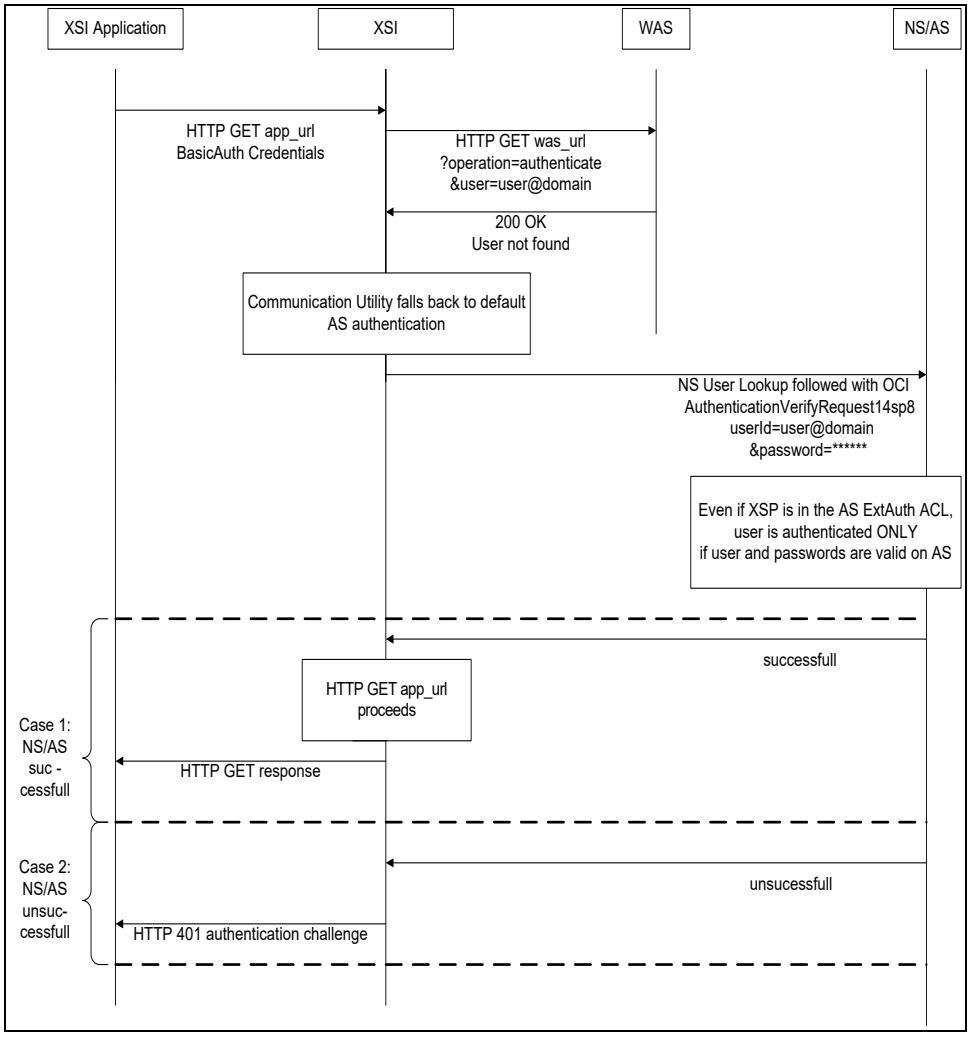

Figure 16 Xtended Services Interface Using External Authentication – User Unknown by WAS with Successful/Unsuccessful Authentication

6.1.7 OCS/OCI-P and OCI-C Interface: Detailing Authentication Using a Web Authentication Server

The Open Client Server on the Xtended Services Platform is able to perform HTTP authentication using a WAS, and the following describes the mechanism behind it.

The Open Client Server intercepts the OCI *AuthenticationRequest* and *LoginRequest* messages coming from the client application, and tries to authenticate the user connection with a Web-based authentication server, instead of proxying them directly to the target Application Server via Network Sever lookup.

The Open Client Server communicates with the WAS using the protocol described in section *6.1.5 Communication Protocol for the Web Authentication Server*.

From the client's perspective, the authentication sequence is divided in two steps:

1) Send an authentication request to OCS to obtain a temporary nonce from the WAS that must be used to build a *Signed Password*.

2) Send the login request to OCS with the *Signed Password* along with optional *plainTextPassword*.

In step1, OpenClientServer sends to the WAS an HTTP authentication request, and obtains a nonce. In step2, OCS forwards to the WAS an HTTP request carrying the *hashedPwd* (that is: client's *Signed Password*) and, provided that configuration parameter *supportsBasicCredentials* is set to "true", the optional *BasicAuth Credentials* derived from the client's provided *plainTextPassword*.

As stated above, even if a customer elects to design a WAS to work with *BasicAuth Credentials* only, the *Signed Password* must be provided to the WAS for protocol requirement purposes, and therefore the WAS must, before all, generate a nonce in its authentication response for the sake of *Signed Password* calculation.

The following figures illustrates the four possible scenarios in OCI-P authentication using a WAS, that is (1) UserId known by WAS – authorized Login, (2) UserId known by WAS – unauthorized Login, (3) UserId unknown by WAS – Authorized Login (defaulting to NS/AS authentication) and (4) UserId unknown by WAS – unauthorized login (defaulting to Cisco BroadWorks Network Server/Application Server authentication).

Note that the OCI-C protocol follows the same scheme described for OCI-P. LoginRequest/LoginResponse and AutenticationRequest/AuthenticationResponse are replaced with RegisterRequest/RegisterResponse and RegisterAuthentication/ResponseAuthentication respectively.

6.1.7.1 OCS External Authentication: UserId known by WAS – Authorized Login

The following figure describes the login result from a client application using OCS for a user successfully authenticated by the WAS and known by Cisco BroadWorks.

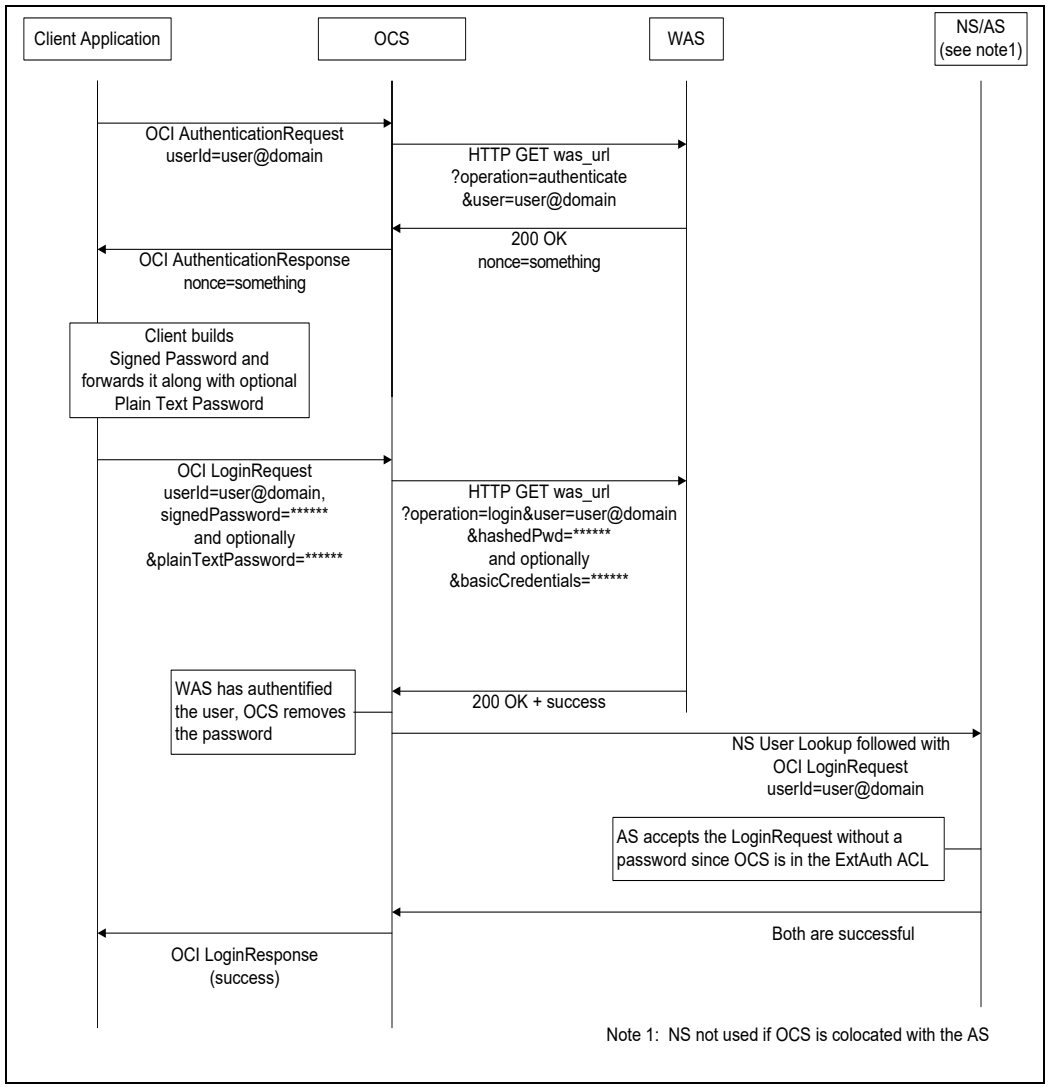

Figure 17 OCS External Authentication – User Known by WAS

6.1.7.2 OCS External Authentication: UserId Known by WAS – Unauthorized Login

The following figure describes the login result from a client application using OCS for a user known by the WAS but unsuccessfully authenticated (bad credentials) (Case1), and a user successfully authenticated by the WAS but unknown by Cisco BroadWorks (Case 2).

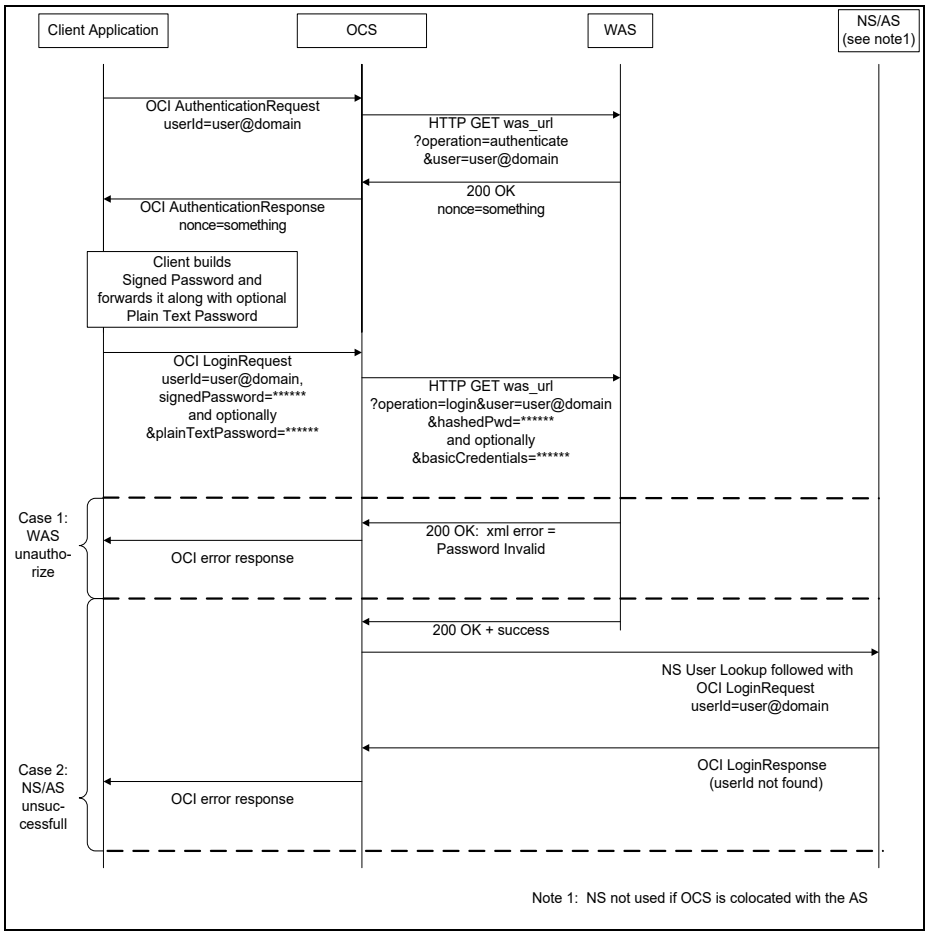

Figure 18 OCS External Authentication – User Known by WAS but Unsuccessfully Authenticated

6.1.7.3 OCS External Authentication: UserId Unknown by WAS - Authorized Login

The following figure describes the login result from a client application using OCS for a user unknown by the WAS but successfully authenticated with a fallback to Network Server/Application Server authentication.

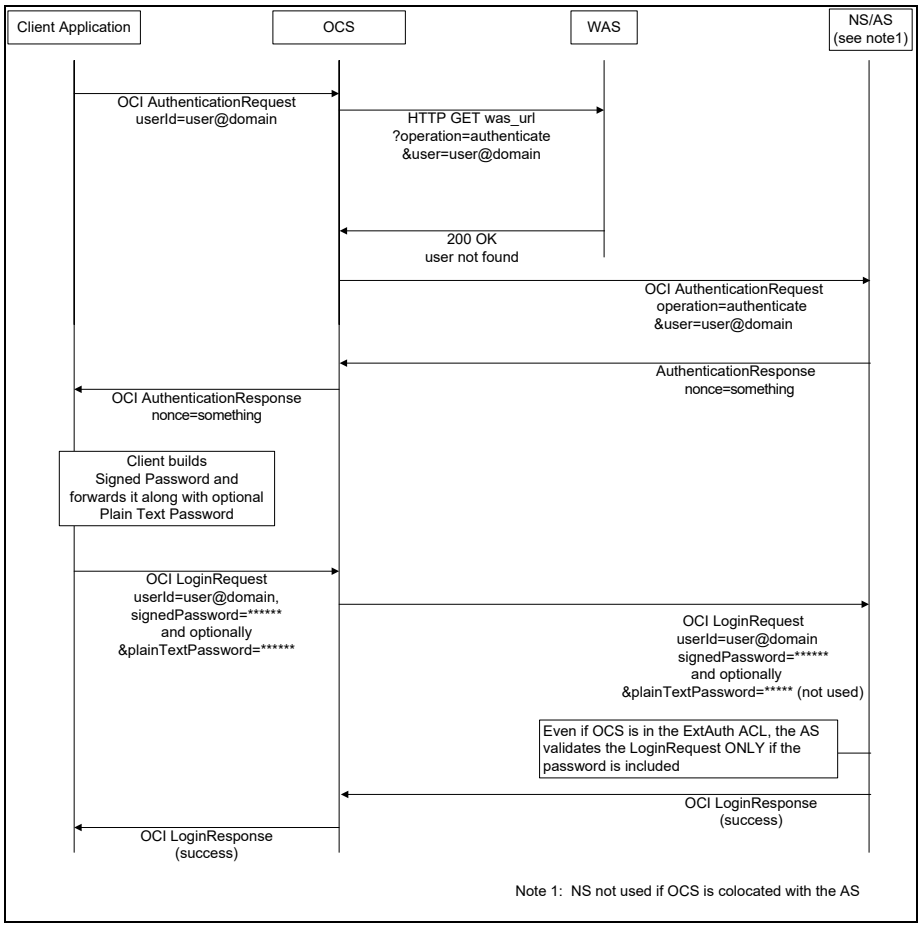

Figure 19 OCS External Authentication – User Unknown by WAS but Successfully Authenticated

6.1.7.4 OCS External Authentication: UserId Unknown by WAS – Unauthorized Authentication/Login

The following figure describes the login result from a client application using OCS for a user unknown by the WAS, with a fallback to Network Server/Application Server authentication, unsuccessfully authenticated, either because of user unknown by Cisco BroadWorks or bad user credentials.

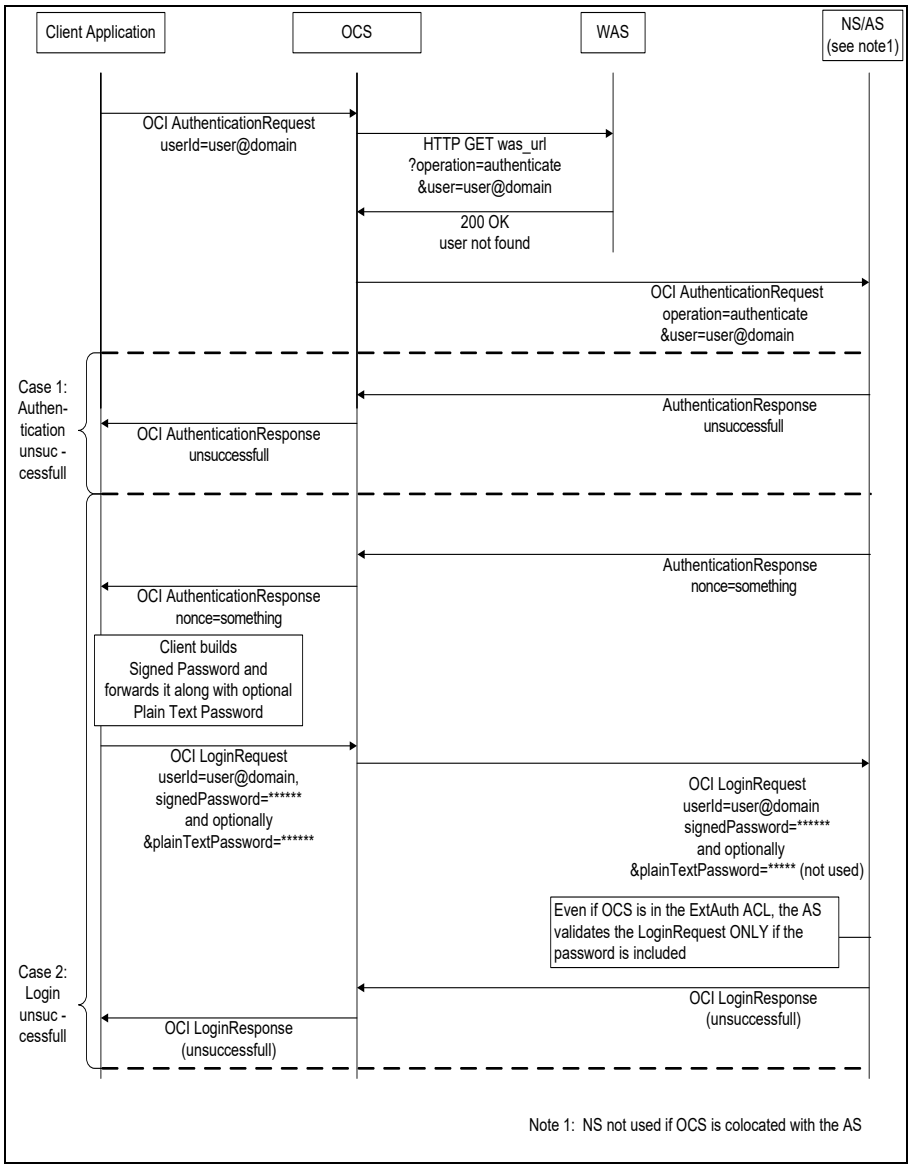

Figure 20 OCS External Authentication – User Unknown by WAS Unsuccessfully Authenticated

6.1.8 WAS Authentication Request and Response Specification

An authentication request is sent by the Xtended Services Interface/Open Client Server to the WAS, using parameters to the URL that are specified by the customer. The following is a list of parameters sent by the Xtended Services Interface/Open Client Server.

Parameter	Possible Values
Operation	“authenticate”
userId	“<userid>@<domain>” as specified in client’s authentication request. The Open Client Server adds the default system domain if no one is specified in the client’s authentication request.

Table 4 WAS Authentication Parameters

Therefore, if the customer provisioned a URL for external authentication, that is *http://server/servlet/authenticationservlet*, the HTTP request would be sent to the WAS as a get request with the following URL.

```
http://server/servlet/authenticationservlet?operation=authenticate&userId
=user@domain
```

The WAS is expected to respond with an XML message in the body of the HTTP response, with the following format.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<com.broadsoft.protocols.extauth.AuthenticationResponse errorCode=”0”
nonce=”58399502945”>
</com.broadsoft.protocols.extauth.AuthenticationResponse>
```

The nonce is returned by the WAS in the AuthenticationResponse. The errorCode is expected to be “0” when the authentication is successful. In case of error, the WAS is expected to return an XML document in the same format, with nonce being set to “empty string” and an error code that is one of the following values.

Error Code	Error
1	User not found.
3	Error processing request. Would be used, for example, if SiteMinder could not be reached, and so on.

Table 5 WAS Authentication Responses

Note that when the WAS is designed to work with *basicAuth Credentials*, a dummy nonce must still be provided by the WAS for protocol requirement purposes. The suggested dummy nonce value is “1234567890”.

6.1.9 WAS Login Request and Response Specification

A login request is sent by the Xtended Services Interface/Open Client Server to the WAS using parameters to the URL that are specified by the customer. The following is a list of parameters sent by the Xtended Services Interface/Open Client Server.

Parameter	Possible Values
operation	"login"
userId	"<userid>@<domain>" as specified in client's authentication request. The Open Client Server adds the default system domain if no one is specified in the client's authentication request.
hashedPwd	The hashed password, as specified in the original login request sent by the client (OCI-P) or calculated by Xsi (either case computed using hashedPwd = MD5(nonce + ":" + SHA (password)).
basicCredentials	The plainTextPassword, as specified in the original login request sent by the client (OCI-P) or carried in URL to the Xtended Services Interface (either case computed to basicCredentials = Base64 (userId + ":" + plainTextPassword) (see note *).

Table 6 WAS Login Parameters

If the customer provisioned a URL for external authentication, that is http://server/servlet/authenticationservlet, the HTTP request would be sent to the WAS as a get request with the following URL.

```
http://server/servlet/authenticationservlet?operation=login&userId=user@d
omain&hashedPwd=f7d483bd91499a340b0dea814d05686c&basicCredentials=
YVVzZXIxQGxvY2FsaG9zdDphdXNlcjE=
```

NOTE: The *basicCredentials* parameter is only included if the configuration parameter *supportsBasicCredentials* is set to "true". In addition, for OCI-P and OCI-C only, the client's parameter *plainTextPassword* must be provided for *basicCredentials* to be included in the previous HTTP URL request.

The WAS is expected to respond with an XML message in the body of the HTTP response, with the following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<com.broadsoft.protocols.extauth.LoginResponse errorCode="0">
</com.broadsoft.protocols.extauth.LoginResponse>
```

The *errorCode* is expected to be "0" when the authentication is successful. In case of error, the WAS is expected to return an XML document in the same format, with an error code that is one of the following values:

Error Code	Error
1	User not found.
2	Invalid password provided.
3	Error processing request. Would be used, for example, if SiteMinder could not be reached, and so on.

Table 7 WAS Login Responses

6.1.10 External Authentication Agent

The Xtended Services Interface can use an external agent to process the authentication process. This agent is also historically called an embedded agent, but it is not required to be collocated on the Xtended Services Platform. Only the HTTP interface of the Xtended Services Interface can use an external authentication agent.

When the external agent functionality is enabled, the Xsi authentication mechanism is bypassed. It is expected that all incoming requests are pre-authenticated by the external agent. The external agent authenticates the user and simply crafts a request to the Xtended Services Platform, adding the user ID in the Xsi request HTTP header.

Special care must be taken to ensure that only requests coming from the external agent can reach the Xsi.

For more information about the embedded agent topology, consult section 5.2.1 External Authentication using Embedded Agent.

6.1.10.1 Custom headers

Custom headers are used to communicate the identity authenticated by the embedded agent to the Xtended Services Interface.

The custom *HTTP* headers expected by the Cisco BroadWorks Xsi are listed in Table 8.

Name	Default Value	Description
HTTP_BW_USERID	None (This header is required.)	The end user's or administrator's user ID in the Cisco BroadWorks system. The value can contain the @domain ending, if required.
HTTP_BW_DOMAIN	""	The part after the "@" sign in the user ID, if required, and if not already provided in the HTTP_BW_USERID header. Specifically, if a "@" is present in the HTTP_BW_USERID header value, the HTTP_BW_DOMAIN value is ignored.

Table 8 Custom HTTP Headers

6.1.10.2 Protect HTTP Interface from Unauthorized Access

When the external authentication agent mechanism is enabled, access to the Xtended Services Platform must be restricted. Note that enabling the external authentication agent does not only affect the Xsi, but also other Web Applications on the Xtended Services Platform.

Enabling the external authentication agent allows any request containing simply a user ID in a header to access the Xsi interface. Therefore, network access to the Xtended Services Platform's HTTP interface must be restricted to only allow connections coming from the external agent.

6.2 Configuration Data for External Authentication

6.2.1 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/HealthCheck

This section describes the *HealthCheck* node that introduces the parameters for enabling the reporting of connection issues with the Authentication Server by initiating a connection attempt at a regular interval using a predefined user.

Name	Type	Content Restrictions	Default Value	Description
healthCheckInterval	seconds	1,3600 nillable	60	This parameter specifies the health check interval. The mechanism is disabled when nil or left blank.
username	token	1,256	BworksHealthCheck	This parameter specifies the username used by the health check mechanism. However, when the selected Authentication Server does specify client authentication data (for example, with LDAP simple), this parameter is ignored.

6.2.2 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/RADIUS

This section describes the RADIUS authentication node that introduces the parameters and sub-nodes documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
authenticationScheme	enumeration	pap, chap, mschapv2	mschapv2	This parameter specifies the RADIUS authentication scheme.
hostname	netAddress	nillable	nil	This parameter specifies the primary IP address, host, or domain name of the RADIUS server.
port	registeredPort	1024,65535	1812	This parameter specifies the RADIUS server port number.
sharedSecret	Base64Password	1,128 nillable	nil	This parameter specifies the shared secret required by all authentication schemes.

6.2.3 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/KERBEROS5

This section describes the configuration required for the Kerberos version 5 authentication node that introduces the parameters documented in the following table. This is also an LDAP SASL security service.

Name	Type	Content Restrictions	Default Value	Description
configurablePrincipal	token		nil	This parameter specifies the configurable principal. It is defined as a string in which {0} is substituted with the user name; a nil value implies {0}. The user name is defined as follows: ▪ In the case of Kerberos version 5 as a Cisco BroadWorks external authentication, {0}:=BWPrincipal (the K5 principal is the Cisco BroadWorks principal). ▪ In the case of LDAP/Kerberos version 5 in direct mode (although not really needed), {0}:=BWPrincipal (the K5 principal is the Cisco BroadWorks principal). ▪ In the case of LDAP/Kerberos version 5 in indirect mode, {0} is the user name obtained from the LDAP search, that is, the same as {0} used in the LDAP *userPattern*.
default realm	token	1,256 nillable	nil	This parameter specifies the default Kerberos realm.
default domain	token			This parameter specifies the default Kerberos domain.
kerberos5KDC	netAddress	1,128 nillable	nil	This parameter specifies the location of the Kerberos key distribution center (KDC) location.
kdcPort	basePort	1,65535	88	This parameter specifies the Kerberos KDC listening port.
ticketTimeout	integer	5,120	8	This parameter specifies the timeout (in seconds) when communicating with the Kerberos KDC.

6.2.4 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP

This section describes the LDAP authentication node that introduces the parameters and sub-nodes documented in the following table. The *userToDnMapping* provides a flexible mechanism for mapping a login user to an LDAP DN. For the *securityAuthentication* parameter, note that the anonymous security authentication does not require any additional configuration. However, the simple and SASL security authentications require an underlying node.

Name	Type	Content Restrictions	Default Value	Description
url	URL	1,255 nillable	nil	This parameter specifies the URL of the external LDAP server. For connecting to the default unsecure LDAP port (389) at broadsoft.com, you can set the url to “ldap://broadsoft.com”. For connecting to the default secure port (636), you can set the URL to “ldaps://broadsoft.com”. Otherwise, you must specify the port (for example, ldap://broadsoft.com:4321).
version	enumeration	2,3	3	This parameter specifies the LDAP version to be used when connecting to the LDAP server.
userToDnMapping	not applicable			This level provides the ability to specify how a login user maps to a LDAP distinguish name.
securityAuthentication	enumeration	anonymous, simple, SASL	anonymous	This parameter specifies the LDAP security authentication.
simple	not applicable			This level provides the ability to view and modify LDAP simple parameters.
sasl	not applicable			This level provides the ability to view and modify SASL parameters.

6.2.4.1 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping

This section describes the LDAP *userToDnMapping* node that introduces the parameters documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
userPattern	string	1,256 nillable	nil	For non-SASL security authentication, this parameter specifies how entries are laid out in the LDAP directory. The keyword "{0}" identifies where the user name substitution occurs. For example, setting the *userPattern* to "uid={0},ou=people,dc=broadsoft, dc=com" associates the *uid* common name with the user name. For SASL security authentication, this parameter specifies which DN attribute to use for authentication. The keyword "{0}" identifies the user name location and must be preceded by the equal sign "=" and the attribute name. For example, after setting the *userPattern* to "cn={0}", the system fetches the value associated with the *cn* attribute and sends it for authentication.
userBase	string	1,256 nillable	nil	This parameter specifies the entry that is the base of the subtree containing users. If it is not specified, the search base is restricted to the top level.
userSearch	string	1,256 nillable	nil	This parameter specifies how to search the LDAP directory. The pattern specifies the LDAP search filter to use after the substitution of the user name.
userSubtree	Boolean		false	This parameter specifies if the search scope is for the top level only ("false") or for the entire subtree ("true").

6.2.4.2 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/simple

This section describes the LDAP simple node that introduces the parameters documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
principal	token	1,256 nillable	nil	This parameter specifies the principal to connect to the LDAP server. This is typically a user name with administrative privileges.
credentials	Base64Password	1,128 nillable	nil	This parameter specifies the credentials to connect to the LDAP server. This is typically the password associated with the principal.

6.2.4.3 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL

This section describes the LDAP SASL node that introduces the parameters and sub-nodes documented in the following table. For the *mechanismName* parameter, note that the DIGEST-MD5 mechanism does not require any additional configuration.

Name	Type	Content Restrictions	Default Value	Description
mechanismName	enumeration	KERBEROS5, DIGEST-MD5	DIGEST-MD5	This parameter specifies the name of the SASL mechanism to use when connecting to the LDAP server.
principal	token	1,256 nillable	nil	This parameter specifies the principal to connect to the LDAP server when a SASL mechanism is used.
credentials	Base64Password	1,128 nillable	nil	This parameter specifies the credentials to connect to the LDAP server when a SASL mechanism is used.
kerberos5	not applicable			This level provides the ability to view and modify the *Kerberos5* Kerberos version 5 SASL parameters.
digest-md5	not applicable			This level provides the ability to view and modify the DIGEST-MD5 SASL parameters.

6.2.4.3.1 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/KERBEROS5

This section describes the Kerberos 5 (also known as GSSAPI) SASL mechanism node that introduces the parameters documented under the stand-alone context for Kerberos 5 in section *6.2.3 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/KERBEROS5*.

6.2.4.3.2 *BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/DIGEST-MD5*

This section describes the DIGEST-MD5 SASL mechanism node that introduces the parameters documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
realm	token	1,256 nillable	nil	This parameter specifies the DIGEST-MD5 realm.

6.2.5 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/WAS

This section describes the WAS node that introduces the parameters documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
enable	Boolean		False	Enables authentication of incoming HTTP requests with an external Web Authentication Server.
url	URL	nillable	nil	URL of the external Web Authentication Server.
timeout	Seconds	5..120	8	Timeout (in seconds) for external authentication with an external Web Authentication Server.
supportsBasic Credentials	Boolean		False	Indicates whether or not the external Web Authentication Server supports the use of *basicCredentials*.

6.2.6 BWCommunicationUtility/DefaultSettings/ExternalAuthentication/EmbeddedAgent

This section describes the Embedded Agent node that introduces the parameters documented in the following table.

Name	Type	Content Restrictions	Default Value	Description
enable	Boolean		False	Allows authentication without credentials.

7 Third-Party System Integration

7.1 External Authentication using Single Sign On

To achieve the specific goal of seamless navigation between a third-party web portal and Cisco BroadWorks web applications, Single Sign-On is implemented. This allows users to navigate transparently (without being asked for additional credentials) or to be redirected from a web application or portal to a Cisco BroadWorks web application.

The Single Sign-On mechanism is available to third-party applications by exposing the required Xtended Services Interface (Xsi) and OCI transactions externally.

7.1.1 Single Sign-On Using Login Tokens – Overview

The Single Sign-On mechanism relies on login tokens. The Single Sign-On sequence goes as follows:

1) Application A is authenticated with the Provisioning Server for a certain user or administrator.
2) Application A requests a login token from the Provisioning Server for the same user or for another user controlled by the administrator. The login token, which is valid for a period of 60 seconds, confirms that the user was authenticated against the authority (its hosting Provisioning Server).
3) Application A launches application B while it communicates/provides the login token to application B.
4) Application B validates the login token with the Provisioning Server as a means to authenticate the user.
5) Upon success, application B is considered *authenticated* against the Provisioning Server, and can therefore interact with this Server as if it had made the authentication process directly with it.

As described in later sections, obtaining a token is done using one of the following:

- A new OCI command: *ExternalAuthenticationCreateLoginTokenRequest*
- A new Xtended Services Interface command: */user/<userid>/profile/loginToken*

Validating the token is done using the existing *AuthenticationVerifyRequest* OCI command.

The following section illustrates the process flow of the Single Sign-On mechanism with Xsi for a Call Center client.

7.1.2 Third-Party Web Portal to Call Center Single Sign-On

The following figure shows the Single Sign-On to the Call Center client application. The same interaction applies to the Receptionist client application as well. Differences specific to the Moderator client application are highlighted.

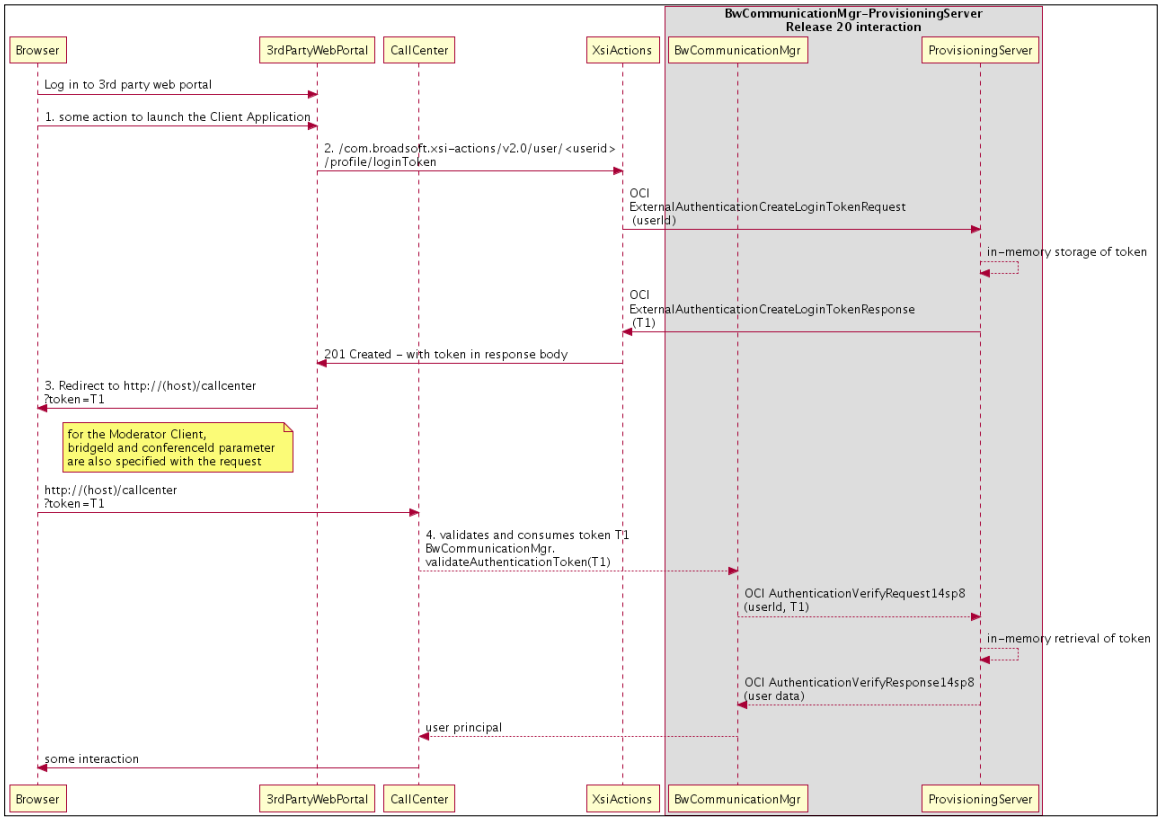

Figure 21 Third-Party Web Portal to Call Center Single Sign-On

The Single Sign-On sequence from a third-party web portal is as follows:

1) From the third-party web portal, the user launches the Call Center application.
2) The third-party web portal uses Xsi-Actions to obtain a login token. Xsi-Actions uses the OCI *ExternalAuthenticationCreateLoginTokenRequest* to create the token on the Provisioning Server for the user. Xsi-Actions sends the token in the body of a 201 (created) HTTP response.
3) The third-party web portal redirects the client (browser) to the Call Center URL. Note that for the Moderator client application, the *bridgeId* and *conferenceId* are included as parameters of the request.
4) The Call Center uses the *BwCommunicationMgr* API to validate the token.
   - *BwCommunicationMgr* uses the OCI *AuthenticationVerifyRequest14sp8* to validate the token.
   - Upon success, the Call Center presents its functionality to the end user.

7.1.3 OCI Application Single Sign-On

The following figure shows how an OCI application can benefit from the Single Sign-On mechanism. It typically does not launch a Cisco BroadWorks web application; however, it can be used to transfer the authentication from one application to another or between different modules of a given application.

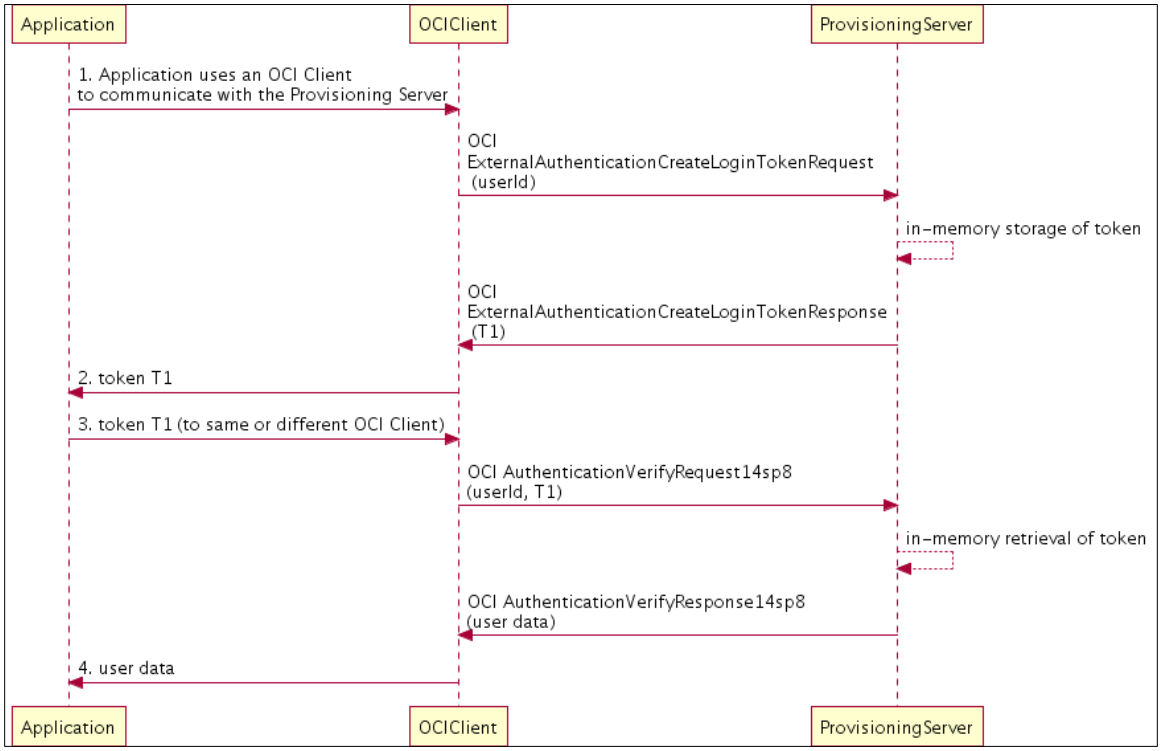

Figure 22 OCI Application Single Sign-On

The Single Sign-On sequence is as follows:

1) An application uses an OCI client to authenticate with the Provisioning Server.
   - The application uses the OCI *ExternalAuthenticationCreateLoginTokenRequest* to obtain login token T1 from the Provisioning Server for the authenticated user.
   - The application communicates login token T1 to a different module or a different application.
2) The application retrieves the login token from the OCI client, using an API call, or any other custom mechanism, to retrieve the login token.
3) The other module or application uses the OCI *AuthenticationVerifyRequest14sp8* (while specifying login token T1) to authenticate as the user with the Provisioning Server.

   The OCI client is then authenticated and can proceed with other OCI commands on behalf of the authenticated user.

7.2 External Authentication Using Network Access Control List

In a third-party system integration scenario, you can bypass the authentication process and automatically accept login requests from trusted hosts using the external authentication network access control list (ACL) of the Open Client Server. In this case, it is the responsibility of the third-party system to make sure users are authenticated. Cisco BroadWorks assumes in this case that incoming requests are trusted from the system hosting the third-party application.

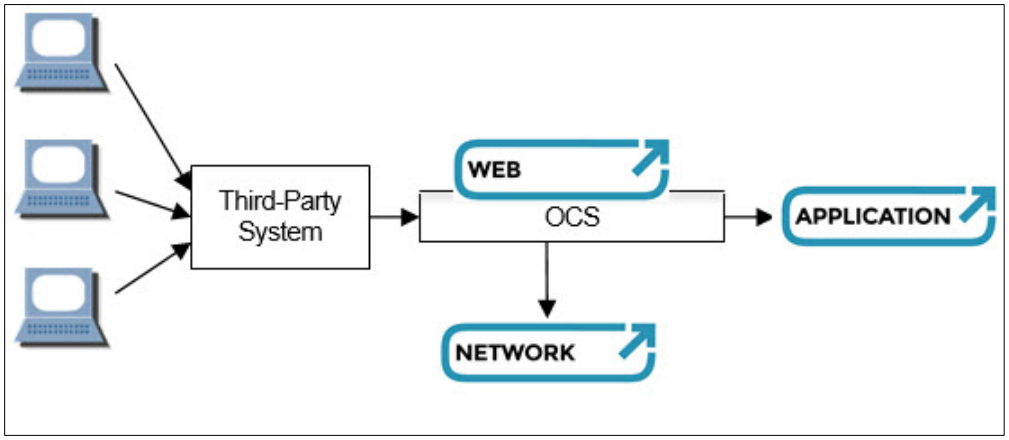

Figure 23 OCI External Authentication for Third-Party Systems

Any trusted host from which you want login requests (OCI-P or OCI-C) without passwords to be accepted automatically must be added to the access list, providing its static IP address. In this case, the authentication request preceding the login request (the transaction used to obtain a nonce to encrypt the password) is not required, since the password does not need to be provided. The session can be established directly by sending a login request.

The ACL is accessible through the CLI. Trusted hosts can be configured using the CLI in the following level.

```
XSP_CLI/Application/OpenClientServer/ExternalAuthentication/AccessControl
List>
```

8 Appendix A: Integration with Netegrity SiteMinder

The following sections describe specific concerns regarding the integration of Cisco BroadWorks Xtended Services Platform and/or collocated Web Server on the Application Server with Netegrity SiteMinder. In addition, there is a description of the custom *HTTP* headers' external authentication mechanism in section *5.2.1 External Authentication using Embedded Agent*.

8.1 Hardware Considerations

The Web Agent is integrated with the Apache Stronghold web server on the Xtended Services Platform that fronts the Application Server machine. If an Xtended Services Platform is not deployed, or if protection is required for the collocated Web Server, then the Web Agent can be installed on the Application Server machine as well. To enforce access control, the Web Agent interacts with the policy server, where all authorization and authentication decisions actually take place.

The policy server can be an existing server in the customer's web portal deployment. It can be installed on the Xtended Services Platform or on a new machine that the customer provides and installs. Installing the policy server on the Application Server machine is not recommended.

8.2 Authentication Process

The Web Agent intercepts all user (browser) requests for resources (HTML files, JSP pages, servlets, graphics, and so on) and checks with the policy server as to whether the requested resource is protected. If the resource is unprotected, the access request proceeds directly to the Xtended Services Platform.

If the resource is protected, the Web Agent determines which authentication method is required to access the requested resource. Typically, the credentials required are a name and password, but the customer can decide to use any SiteMinder-supported authentication scheme. The Web Agent challenges the users for their credentials, according to the authentication method required by the policy server. The users then respond with the appropriate credentials. The Web Agent passes these credentials back to the policy server, which uses them to authenticate the users against a user directory.

8.3 Authorization Process

When a user attempts to access a protected resource, the policy server first authenticates the user, as previously explained. Users are then authorized to access resources based on policies. The Web Agent can also forward additional user-specific attributes, in particular the user ID in the Cisco BroadWorks system, to the web application in the form of a response. A user is authorized as follows:

- The SiteMinder agent sends the details of the HTTP request along with the user's identity to the policy server for authorization.
- The policy server determines which policies protect the resource in question and whether the policies apply to the user attempting access.
- The policy server communicates its decision to grant or deny user access along with the applicable responses to the agent.
- If the policy server grants access, the agent adds the user-specific attributes to the *HTTP* header, which is then forwarded to the Application Server for processing.

8.4 Policy Server Installation

The policy server installation can be an existing installation shared with other parts of the customer's web portal, or a new installation specifically for the Cisco BroadWorks web application. Connecting to an existing installation would be preferable, because it requires less hardware resources and less maintenance.

Whether an existing policy server is used, or a new one is installed, the same policies must be created to protect the Cisco BroadWorks web application. The policy contains rules, which are assigned against a realm, which is a collection of resources. The policy required to protect a Cisco BroadWorks installation can be simple, with one realm that covers the entire application. The complexity is in connecting to a user repository and obtaining the user ID (and domain, if required) that must be passed to the Cisco BroadWorks web application as custom *HTTP* headers. When it is created, a policy is assigned against agents, or more precisely, against "agent identities", as an agent can have multiple identities. For a Cisco BroadWorks installation, the agents have a single identity, and agents within the same cluster share the same policy.

8.5 Custom HTTP Headers

The *ea_login.jsp JSP* page is expecting the user ID and domain information in the form of custom *HTTP* headers, as shown in Table 1.

To authenticate the end user or administrator and obtain the custom *HTTP* header information, the policy server must connect to the customer's user data repository, such as a LDAP directory server, NT domain, Open Database Connectivity (ODBC) database, and so on. This is a customer-specific procedure that requires custom development work.

Note that the custom *HTTP* headers are only needed for the *ea_login.jsp* page. Therefore, the realm of the Cisco BroadWorks web application can contain two rules: one covering everything with no Response, and another covering the *ea_login.jsp* page with a Response providing the custom *HTTP* headers. An additional rule may be needed to block the */Login* page, as described in section *5.2.3.2 Unauthorized Login Attempts through Cisco BroadWorks Login Page*.

8.6 Sharing Policy Server Polices Across Clusters

Depending on customer preferences, it might be possible to reuse the same policy server and even the same policy for multiple Application Server clusters.

8.7 Policy Server High Availability

For both installation scenarios, that is, reusing an existing policy server installation or having a new policy server installation, it is assumed that policy servers are redundant (to satisfy general Cisco BroadWorks high availability requirements). In accordance with the way high availability is configured in the remainder of the Cisco BroadWorks system, it is recommended to have the policy servers in failover mode, rather than round-robin mode.

8.8 Web Agent Installation

The Web Agent is installed on the same server as the Xtended Services Platform. If protection is needed for the collocated Web Server on the Application Server, another Web Agent instance must be installed on the Application Server machine. The version of the policy server and the Web Agent depends on compatibility with the SiteMinder version used in the rest of the customer's site. In addition, a Web Agent must be available for the Xtended Services Platform used by the current release of Cisco BroadWorks.

8.9 Single Sign-On Configuration

Depending on the choice of domain names in the remainder of the customer's web portal, the Web Agent should be installed so as to support Single Sign-On either within one domain or across multiple domains. If Single Sign-On spans multiple domains, a specially configured SiteMinder Agent serves as a "cookie provider". The customer is expected to set up the cookie provider Agent.

According to the SiteMinder documentation, when using replicated user directories with non-replicated policy stores in a Single Sign-On setup, the user directory must be named identically for all policy stores.

In addition, the policies established to protect the Cisco BroadWorks web application must not require a higher level of protection than the realm the end user or administrator is coming from elsewhere in the customer's portal. Otherwise the Web Agent forces the end user or administrator to re-enter their credentials upon accessing the Cisco BroadWorks realm.

8.10 Agent Key Management

Web Agents use agent keys to encrypt and decrypt all SiteMinder cookies. The agent uses the agent key to encrypt cookies before sending them to a user's browser and to decrypt cookies when it receives them from other Web Agents. All Web Agents need to be aware of the same keys, and the keys must be set to the same value for all agents' communication with a policy server. This is particularly important for agents in a Single Sign-On environment, which is preferred for a Cisco BroadWorks setup.

Therefore, agent key management must be integrated with other SiteMinder components in the customer's web portal, according to the *SiteMinder Policy Server Operations Guide*. The specifics are dependent on the SiteMinder release number.

8.11 Provisioning End Users and Administrators

It is assumed that the SiteMinder Web Agent instances on the Xtended Services Platform s that front an Application Server cluster connect to an existing SiteMinder policy server that contains profiles of Cisco BroadWorks users (end users and administrators). This is also known as a Single Sign-On configuration.

For each end user and administrator known to the policy server, a corresponding entity must be created in the Cisco BroadWorks system, for instance, through the OCI API. These entities are created without specifying a password, as they are accessing the Cisco BroadWorks web application through the Web Agent-protected Xtended Services Platform that fronts the Application Server.

8.12 Levels of Protection

The Web Agents are configured to offer the same level of protection for the JSP pages and servlets in the Cisco BroadWorks web application, regardless of the user type (end user or administrator). Once a Cisco BroadWorks end user or administrator accesses the entry point into the web application, further authorization is performed on a per-page basis based on the user's authorization level in the Cisco BroadWorks system. This simplifies the setup of the Web Agents, while providing an adequate level of protection.

For an increased level of protection, the different types of users in Cisco BroadWorks would need different policies in the policy server. When this redundant level of protection is required, Cisco provides filters to set up the rules for the different policies (end user, department administrator, and so on), which would probably be in the form of hierarchical realms.

Finally, the web login authorization level protection is still in place. Both the Application Server and Xtended Services Platform CommPilot applications support login access control, providing the ability to restrict access to a specific user level and levels below that. This control can be configured via the Cisco BroadWorks command line interface level *AS_CLI/Applications/CommPilot/GeneralSettings*. Usually, a group administrator is the highest level of administrator for which external web access should be allowed.

8.13 Session Management

The policy server component ensures session management, including session duration tracking. The Cisco BroadWorks web application web session duration tracking mechanism does not apply to end users, department, group, or service provider administrators when external authentication password rules are enabled.

8.14 Cross-site Scripting and Escaped Characters in URLs

By default, SiteMinder is set to block any URLs that contain escaped characters. For example, the "<" character can usually be escaped by substituting "%3C". However, because such escaped characters can be used in cross-site scripting attacks to embed HTML tags in URLs, SiteMinder does not allow access to any URL that contains them.

You can configure the SiteMinder Web Agent to disable cross-site scripting prevention by setting its *CSSChecking* parameter to "no". For more information on configuring the Web Agent, see the SiteMinder documentation.

9 Appendix B: External Authentication with LDAP Examples

9.1 Basic User Pattern Example

9.1.1 LDAP Directory Elements

Following are the basic directory elements that capture how to connect to the LDAP server and the list of known users along with their characteristics. This example represents the LDAP server-side configuration that is expected to be already present in the Cisco customer network.

- The root naming context and authentication user:

  # top-level entry

  dn: dc=broadsoft,dc=com
  objectClass: dcObject
  dc:broadsoft

  # authentication

  dn: dc=broadsoft,dc=com,dc=auth
  objectClass: dcObject
  dc:auth

  # SASL DN

  uid=bwadmin@broadsoft.com,cn=digest-md5,cn=auth

- The end users (Jane Doe and John Smith):

  # define an entry to contain people
  # searches for users are based on this entry
  dn: ou=people,dc=broadsoft,dc=com
  objectClass: organizationalUnit
  ou: people

  # define a user entry for Jane Doe

  dn: uid=jdoe@broadsoft.com,ou=people,dc=broadsoft,dc=com
  objectClass: inetOrgPerson
  uid: jdoe
  sn: jane
  cn: jane doe
  mail: j.doe@broadsoft.com
  userPassword: jane

  # define a user entry for John Smith

  dn: uid=jsmith@broadsoft.com,ou=people,dc=broadsoft,dc=com
  objectClass: inetOrgPerson
  uid: jsmith
  sn: jsmith
  cn: john smith
  mail: j.smith@broadsoft.com
  userPassword: john

9.1.2 Cisco BroadWorks Configuration

The following data captures the relevant CLI levels for connecting to the LDAP server using the SASL DIGEST-MD5 mechanism.

The following example shows where the distinguished name of the user's entry contains the user name (in the form of an e-mail address) presented for authentication but is otherwise the same for all users. In these cases, the configuration uses the *userPattern* parameter. This shows the Cisco BroadWorks configuration for LDAP (basic user pattern).

```
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> get
  authenticationType = ldap
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> LDAP
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> get
  url = ldap://broadsoft.com
  version = 3
  securityAuthentication = SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> userToDnMapping
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping>
get
  userPattern = uid={0},ou=people,dc=broadsoft,dc=com
  userBase = nil
  userSearch = nil
  userSubtree = false
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping>q
; SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL> get
  mechanismName = DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL> DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/DIGEST-MD5>
get
  Secret = bwadmin
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/DIGEST-
MD5>quit
```

9.1.3 Bind

This section describes the use of the SASL DIGEST-MD5 mechanism using secrets stored directly in the LDAP directory. Take note that other store options exist on the server. DIGEST-MD5 relies on the client and the server sharing a "secret", usually a password. The server generates a challenge and the client generates a response proving that it knows the shared secret.

Based on the Cisco BroadWorks configuration described in the example in section *9.1.2 Cisco BroadWorks Configuration*, the Xtended Services Platform connects to the LDAP server using the URL parameter *ldap://broadsoft.com*. Once the Xtended Services Platform is authenticated as an LDAP client based on the shared secret value, the Xtended Services Platform determines the user's distinguished name by substituting the user name into the *userPattern*, and confirms its identity by binding to the directory with this DN and the password received from the user.

In other words, when Jane Doe logs in using jdoe@broadsoft.com, the following substitution occurs before binding to the LDAP server:

```
Original user_pattern uid={0},ou=people,dc=broadsoft,dc=com
Substituted user_pattern
uid=jdoe@broadsoft.com,ou=people,dc=broadsoft,dc=com
```

Authentication is confirmed once the LDAP server successfully compares the received hash password with its local value.

9.2 User Search Pattern Example

9.2.1 LDAP Directory Elements

Following are the basic directory elements that capture how to connect to the LDAP server and the list of known users along with their characteristics. This example represents the LDAP server-side configuration that is expected to be already present in the Cisco customer network.

- The root naming context and authentication user (DIGEST-MD5)

  # top-level entry

  dn: dc=broadsoft,dc=com
  objectClass: dcObject
  dc:broadsoft

  # authentication

  dn: dc=broadsoft,dc=com,dc=auth
  objectClass: dcObject
  dc:auth

  # SASL DN

  uid=bwadmin@broadsoft.com,cn=digest-md5,cn=auth

- The end users (Jane Doe and John Smith)

  # define an entry to contain people
  # searches for users are based on this entry
  dn: ou=people,dc=broadsoft,dc=com
  objectClass: organizationalUnit
  ou: people

  # define a user entry for Jane Doe

  dn: uid=jdoe@broadsoft.com,ou=people,dc=broadsoft,dc=com
  objectClass: inetOrgPerson
  uid: jdoe
  sn: jane
  cn: jane doe
  mail: j.doe@broadsoft.com
  userPassword: jane

  # define a user entry for John Smith

  dn: uid=jsmith@broadsoft.com,ou=people,dc=broadsoft,dc=com
  objectClass: inetOrgPerson
  uid: jsmith
  sn: jsmith
  cn: john smith
  mail: j.smith@broadsoft.com
  userPassword: john

9.2.2 LDAP Simple

9.2.2.1 Cisco BroadWorks Configuration

The following data captures the relevant CLI levels for connecting to the LDAP server using the LDAP simple mechanism.

The following example shows where the LDAP server must search the directory to find a unique entry containing the user name. In these cases, the configuration uses a combination of the *userBase* and *userSearch* parameters.

```
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> get
  authenticationType = ldap
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> LDAP
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> get
  url = ldap://broadsoft.com
  version = 3
  securityAuthentication = simple
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> userToDnMapping

BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping>
get
  userPattern = nil or uid={0},ou=people,dc=broadsoft,dc=com
  userBase = ou=people,dc=broadsoft,dc=com
  userSearch = (mail={0})
  userSubtree = false
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping>
```

9.2.2.2 Bind

This section describes the use of the LDAP simple mechanism for searching the LDAP database.

Assume that the users still expect to enter their e-mail address (rather than their user ID) when logging in. In this case, the Xtended Services Platform must search the directory for the user's entry. When John Smith logs in as "j.smith@broadsoft.com", the Xtended Services Platform searches the directory for a unique entry with that value as its mail attribute.

```
Original user_search (mail={0})
Substituted user_search (mail= j.smith@broadsoft.com)
```

Once the Xtended Services Platform obtains a unique match from the LDAP server, it must create the DN. If the *userPattern* is "nil", it uses the default DN provided in the search result. Otherwise, it performs the mapping shown in the following example.

```
Original userPattern (uid={0},ou=people,dc=broadsoft,dc=com)
Substituted userPattern (uid=jsmith,ou=people,dc=broadsoft,dc=com)
```

Then, it binds to the directory as uid=jsmith,ou=people,dc=broadsoft,dc=com with the given password. Authentication is confirmed once the LDAP server successfully compares the received password with its local value.

9.2.3 LDAP SASL Digest-MD5

9.2.3.1 Cisco BroadWorks Configuration

The following data capture the relevant CLI levels for connecting to the LDAP server using the SASL DIGEST-MD5 mechanism.

The following example shows where the LDAP server must search the directory to find a unique entry containing the user name. In these cases, the configuration uses a combination of the *userBase* and *userSearch* parameters.

```
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> get
  authenticationType = ldap
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> LDAP
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> get
  url = ldap://broadsoft.com
  version = 3
  securityAuthentication = SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP>
userToDnMapping
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userTo
DnMapping> get
  userPattern = nil or uid={0}
  userBase = ou=people,dc=broadsoft,dc=com
  userSearch = (mail={0})
  userSubtree = false
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userTo
DnMapping>q; SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL>
get
  mechanismName = DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL>
DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/D
IGEST-MD5> get
  Secret = bwadmin
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/D
IGEST-MD5>quit
```

9.2.3.2 Bind

This section describes the use of the SASL DIGEST-MD5 mechanism for searching the LDAP database.

Assume that the users still expect to enter their e-mail address (rather than their user ID) when logging in. When John Smith logs in as “j.smith@broadsoft.com”, the Xtended Services Platform follows one of the following two scenarios:

- If the *userSearch* and the *userPattern* are “nil”, the Xtended Services Platform uses the login user value “j.smith@broadsoft.com” directly as the SASL user.
- If the *userSearch* is configured, the Xtended Services Platform must search the directory for the user’s entry. The Xtended Services Platform searches the directory for a unique entry with that value as its mail attribute.

```
Original user_search (mail={0})
Substituted user_search (mail= j.smith@broadsoft.com)
```

Once the Xtended Services Platform obtains a unique match from the LDAP server, it must create the SASL user.

If the *userPattern* is “nil”, it uses the value associated with the *uid* attribute or the *cn* attribute (if the *uid* attribute is not available). Hence, the Xtended Services Platform uses *jsmith* as the SASL user.

Otherwise, it performs the mapping shown in the following example.

```
Original userPattern uid={0}
Substituted userPattern jsmith
```

Then, it binds to the directory as *jsmith* with the given password. Authentication is confirmed once the LDAP server successfully compares the received password with its local value.

9.3 Redundant LDAP Servers Example

9.3.1 LDAP SASL Digest-MD5

9.3.1.1 Cisco BroadWorks Configuration

The following data captures the relevant CLI contexts for connecting to redundant LDAP servers using the SASL digest-MD5 mechanism.

The following example shows where the URL must be configured to perform SRV lookups.

```
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> get
  authenticationType = ldap
BWCommunicationUtility/DefaultSettings/ExternalAuthentication> LDAP
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> get
  url = ldap://ldap.broadsoft.com
  version = 3
  securityAuthentication = SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP> userToDnMapping
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping> get
  userPattern = nil or uid={0}
  userBase = ou=people,dc=broadsoft,dc=com
  userSearch = (mail={0})
  userSubtree = false
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/userToDnMapping>q; SASL
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL> get
  mechanismName = DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL> DIGEST-MD5
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/DIGEST-MD5> get
  Secret = bwadmin
BWCommunicationUtility/DefaultSettings/ExternalAuthentication/LDAP/SASL/DIGEST-MD5>quit
```

The following example shows an excerpt from the *namedefs* file that contains SRV records to define redundant LDAP servers. As with configured URLs, the SRV targets must correspond to the respective server’s service principal name for the server to accept the connection.

```
_ldap._tcp.ldap.broadsoft.com SRV 1 10 1234 ldap1.broadsoft.com
_ldap._tcp.ldap.broadsoft.com SRV 2 10 4321 ldap2.broadsoft.com
```

9.3.1.2 Bind

This section describes the use of binding to redundant servers.

The Xtended Services Platform first performs a SRV lookup using the configured hostname of the URL as a domain. This results in the following URLs.

```
ldap://ldap1.broadsoft.com:1234
```

```
ldap://ldap2.broadsoft.com:4321
```

The first URL is attempted first since it has a higher priority. A bind is attempted according to the current search and pattern configuration. If the bind fails because of authentication issues, the whole bind operation fails immediately and the other redundant servers are not attempted. If the first server experiences communication issues, a bind is attempted on the second server. Authentication is confirmed once the LDAP server successfully compares the received password with its local value.

10 Appendix C: How to Enable RADIUS or LDAP from MS Active Directory

This section describes which server role of the MS Active Directory must be enabled to support RADIUS or LDAP. A later release of this document is to provide complete documentation of the expected configuration.

NOTE 1: The MS Active Directory server may need to be restarted after installation completes.

NOTE 2: The installation time may be significant. It is important to plan accordingly.

NOTE 3: Administrative privileges are required to perform most actions described in this section.

NOTE 4: The steps described in this section are sufficient for enabling the associated authentication mechanism. However, this list is not exhaustive and other approaches using different configurations can achieve the same result.

10.1 RADIUS

For RADIUS support, you must enable the Network Policy Server (NPS) role from the MS Active Directory. The NPS role may be enabled from the Server Manager.

1) To open the Server Manager, click **Start**, and then click **Administrative Tools**
2) Click **Server Manager**.
3) From the left navigation tree of the Server Manager, click **Roles**.
4) From the right panel of the Server Manager, click **Add Roles**.
5) From the Add Roles Wizard:
   a. Click **Next**.
   b. Click **Network Policy and Access Services**.
   c. Click **Next**.
   d. Select *Network Policy Server* and *Routing and Remote Access Services*.
   e. Click **Next**.
   f. Click **Install** (installation time may be significant).
   g. Click **Close**.

10.1.1 RADIUS Clients

The NPS role within MS Active Directory uses an access list for client requests. Hence, you must have a list of RADIUS clients that correspond to the Cisco BroadWorks servers (for example, Xtended Services Platforms) that issue RADIUS access request messages. The configuration of a RADIUS client includes the value for the shared secret that also must be configured on the Cisco BroadWorks CLI under the *XSP_CLI/System/CommunicationUtility/DefaultSettings/ExternalAuthentication/RADIUS* level.

A RADIUS client can be added from the Server Manager.

1) To open the Server Manager, click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2) From the left navigation tree of the Server Manager, click **Roles** and expand it.
3) Click **Network Policy and Access Services** and expand it.
4) Right-click **RADIUS Clients** and select *New RADIUS client*.
5) Specify the *Friendly name*, the *Address*, and the *Shared secret*. The other fields can retain the default values.
6) Click **OK**.

Repeat the previous steps for every Cisco BroadWorks server that is expected to issue RADIUS access request messages.

10.1.2 Network Policies

The NPS role within MS Active Directory relies on network policies for granting access to users. You must confirm that at least one network policy matches the proper conditions to grant access to Cisco BroadWorks end users. In addition, the policy must have the relevant processing order to avoid being invalidated by another preceding policy that would deny access to the same Cisco BroadWorks end users.

A network policy can be added from the Server Manager.

1) To open the Server Manager, click **Start**, click **Administrative Tools**, and then click **Server Manager.**
2) From the left navigation tree of the Server Manager, click **Roles** and expand it.
3) Click **Network Policy and Access Services** and expand it.
4) Click **NPS (local)** and expand it.
5) Click **Policies** and expand it.
6) Follow the steps from the wizard according to your needs.

   You may need to change the processing order of the newly created network policy.

   a. From the right panel, right-click on the newly created network policy.
   b. Select *Move Up* or *Move Down* according to your needs.

NOTE: You may have to use the main administrator account of Windows Server to view the *NPS (local)* level.

10.2 LDAP

For LDAP support, you must enable the Active Directory Domain Services (ADDS) server role from the MS Active Directory. The ADDS role may be enabled from the Server Manager.

1) To open the Server Manager, click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2) From the left navigation tree of the Server Manager, click **Roles**.
3) From the right panel of the Server Manager, click **Add Roles**.

4) From the Add Roles Wizard:

   a. Click **Next**.

   b. Click **Active Directory Domain Services**.

   c. Click **Next**.

   d. Click **Next**.

   e. Click **Install**. Note that the installation time may be significant.

   f. Click **Close**.

10.2.1 LDAP User Principal Name Suffixes

Each user account in the Active Directory has a user principal name (UPN) based on *RFC 822, Standard for the Format of ARPA Internet Text Messages*. The UPN is composed of the user login name and the UPN suffix joined by the @ sign.

Under certain circumstances, users who need authentication may have a different user principal name suffix. The MS Active Directory provides a mechanism to allow authentication for specific users with a different suffix. For example, if your organization uses a deep domain tree, organized by department and region, domain names can be quite long. The default user UPN for a user in that domain may be sales.westcoast.microsoft.com. The login name for a user in that domain would be user@sales.westcoast.microsoft.com. Creating a UPN suffix of *microsoft.com* would allow that same user to log in using the much simpler login name of user@microsoft.com.

More specifically, to add UPN suffixes:

1) Open Active Directory Domains and Trusts.

2) In the console tree, right-click **Active Directory Domains and Trusts**, and then click **Properties**.

3) On the *UPN Suffixes* tab, type an alternative UPN suffix for the forest, and then click **Add**.

Repeat step 3) to add additional alternative UPN suffixes.

10.2.2 LDAP Digest-MD5

The support for LDAP Digest-MD5 requires the following configuration changes that are not performed by default on the MS Active Directory system.

The support for digest authentication may be enabled by adding the server role named Web Server (Internet Information Services [IIS]).

Alternatively, you can enable the property named *Store passwords by using reversible encryption,* which affects the digest hashes. You need to access the Server Manager to perform this configuration change.

1) To open the Server Manager, click **Start**, click **Administrative Tools**, and then click **Active Directory Users and Computers**.

2) From the left navigation tree of the Active Directory Users and Computers:

   a. Select the domain name and expand it.

   b. Click **Users**.

3) From the right panel of the Server Manager, double-click on the selected user.

4) From the user *Properties* window:

   a. Select the Account tab.

5) From the *Account* option pane:

   a. Click **Store passwords by using reversible encryption**.

   b. Click **OK** or **Apply**.

Repeat the previous steps to enable Digest-MD5 for additional users.

Once this property is enabled, the user password **must** be reset to force the storing of the password again using the reversible encryption method.

10.2.3 LDAP SSL

The support for LDAP Secure Socket Layer (SSL) requires the Active Directory Certificate Services (ADCS) server role from the MS Active Directory. The ADCS role may be enabled from the Server Manager.

1) To open the Server Manager, click **Start**, click **Administrative Tools**, and then click **Server Manager**.

2) From the left navigation tree of the Server Manager, click **Roles**.

3) From the right panel of the Server Manager, click **Add Roles**.

4) From the Add Roles Wizard:

   a. Click **Next**.

   b. Click **Active Directory Certificate Services**.

   c. Click **Next**.

   d. Click **Next**.

   e. Select *Certification Authority*.

   f. Click **Next**.

   g. Select *Enterprise* (or the setup type accordingly with your profile).

   h. Click **Next**.

   i. Select *Root CA* (or the CA type accordingly with your profile).

   j. Click **Next**.

   k. Select *Create a new private key* (or the private key accordingly with your profile).

   l. Click **Next**.

   m. Select *Cryptography for CA* accordingly with your profile.

   n. Click **Next**.

   o. Configure the CA name accordingly with your profile.

   p. Click **Next**.

   q. Configure the validity period accordingly with your profile.

   r. Click **Next**.

   s. Configure the certificate database accordingly with your profile.

   t. Click **Next**.

u. Review the *Confirmation* page.

v. Click **Install** (installation time may be significant).

w. Click **Close**.

10.3 Kerberos 5

For Kerberos 5 support, a mapping from the Kerberos principals to Windows accounts must be made explicitly. This support requires a set of manual commands from a DOS prompt.

The example data in the following table is used in the instructions that come after it.

Kerberos 5 Realm	**broadsoft.com**
Domain	BROADSOFT.COM
Kerberos 5 Key Distribution Center (KDC)	hostForKdc.broadsoft.com

1) To open a DOS prompt, click **Start** and then click **Run**.

2) From the *Run* window, write "cmd" and click **OK**.

3) From the DOS prompt, use the following commands:

   a. To enable Kerberos authentication and provide a mapping from Kerberos principals to Windows accounts:

      i. c:\ ksetup /AddKdc BROADSOFT.COM

      ii. c:\ ksetup /AddKdc BROADSOFT.COM hostForKdc.broadsoft.com

      iii. c:\ ksetup /MapUser * *

   b. To confirm the mapping:

      i. c:\ ksetup /DumpState

         default realm = broadsoft.com (NT Domain)

         Mapping all users (*) to a local account by the same name (*)

   c. Use the following commands to allow Kerberos server host identification (otherwise, it throws a *Server not found in Kerberos database*)

      i. c:\ setspn -A ldap/ hostForKdc

         Registering ServicePrincipalNames for
         CN= hostForKdc,OU=Domain Controllers,DC=broadsoft,DC=com

         ldap/ hostForKdc

         Updated object

      ii. c:\ ksetup /Domain

         Using domain BROADSOFT.COM.

         default realm = broadsoft.com (NT Domain)

         BROADSOFT.COM:

         – kdc = hostForKdc.broadsoft.com

         – Realm Flags = 0x0No Realm Flags

         Mapping all users (*) to a local account by the same name (*)

11 Acronyms and Abbreviations

ACL	Access Control List
ADCS	Active Directory Certificate Services
ADDS	Active Directory Domain Services
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
AS	Application Server
BW	BroadWorks
CHAP	Challenge-Handshake Authentication Protocol
CLI	Command Line Interface
CN	Common Name
DC	Domain Component
DN	Distinguished Name
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
GSSAPI	Generic Security Services Application Program Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure Sockets
IP	Internet Protocol
JSP	Java Server Page
KDC	Key Distribution Center
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Over SSL
LDIF	LDAP Data Interchange Format
MD5	Message Digest 5 Algorithm
MMTel	Multimedia Telephony
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol
NPS	Network Policy Server
NS	Network Server
OCI	Open Client Interface
OCI-P	Open Client Interface-Provisioning
OCS	Open Client Server
ODBC	Open Database Connectivity
OSS	Operations Support System
OU	Organizational Unit

PAP	Password Authentication Protocol
RADIUS	Remote Authentication Dial-In User Service
RAS	Remote Access Server
SASL	Simple Authentication and Security Layer
SNMP	Simple Network Management Protocol
SRV	Service Record
SSL	Secure Sockets Layer
SSO	Single Sign-On
TCP	Transmission Control Protocol
UPN	User Principal Name
URL	Uniform Resource Locator
WAS	Web-based Authentication Server
Xsi	Xtended Services Interface
Xsp	Xtended Services Platform