FORUM SYSTEMS SENTRY™ VERSION 9.1 ACCESS CONTROL GUIDE

Forum Systems Sentry™ Version 9.1 Access Control Guide, published July 2020.

D-ASF-SE-018832

Table of Contents

INTRODUCTION TO THE ACCESS CONTROL GUIDE

Conventions Used in the Access Control Guide

A red asterisk (*) aligned with a field term means that this field is required. In this and other documentation, the Web Administration Interface is referred to as the WebAdmin and the Forum XML Security Appliance™ is referred to as the 'device', 'product' or 'system'.

In this document, all data or commands that must be entered or selected are displayed in boldface. Example:

User name: **johnsmith**
Password: **********

UI screens which display a STATUS column represent the following states:

- Green status light = enabled policy.
- Yellow status light = a required functional element of this policy is disabled.
- Red status light = disabled policy.

Throughout this and other documents in the Documentation Set, repetitive actions such as:

- View / edit a policy.
- Enable / disable a policy.
- Delete a policy.
- Rename a policy.
- Limit display of policies with Search or Max Results fields.

are not shown. For more information, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9.1 Web-based Administration Guide.*

Customers with plug-in licenses on the system, confirm that your Integration License is visible on the General Info screen under the SUPPORTED FEATURES section.

Any specifications and constraints referenced in this volume appear in the Appendices of this volume.

ACCESS CONTROL AND SECURITY OPTIONS IN THE SYSTEM

Access Control and Authentication

Access Control and Authentication manage security processing in the Sentry in two different ways:

- By applying security at the client side to offload security processing for outbound transactions.
- By applying security at the server side to enforce security policies and protect back-end resources.

Access Control is the process of granting access to authenticated users to system resources and security web services defined in policies. Access Control includes:

- Client side security
- Server side security

Client Side Security

When a client machine sends a document to Sentry, security attributes on an inbound document are validated and policies are applied to enforce security decisions. When the user has been identified, the document is then processed according to the configured policy.

Server Side Security

With outbound documents, the user is identified and authenticated, security attributes on outbound documents are validated and policies are applied to enforce security decisions.

Authentication

Authentication is the process of identifying an individual, organization or machine through usernames, passwords, tokens and smart cards. The use is identified from the protocol using a standard, such as HTTP Basic Authentication, or from the message itself via a SAML Assertion or a WS-Security header.

The product supports four categories of authentication:

- WS-Security SOAP Header Tokens
- SAML Assertions
- HTTP Based Authentication Options
- SSL X509 Client Authentication

Access control and authorization on the product is unique because the product supports transport-centric as well as document-centric mechanisms, where transport-centric examples are HTTP Basic Authentication and SSL Client Certificates and document-centric authentication examples are SAML and WS-Security header credentials, such as Username Token, X.509 Binary Token, X.509 Distinguished Name Token, and Email Token.

Credential Binding Options

The following graphic displays some of the available credential binding options and how they interact with Network Policies. This is a sample scenario and not all access control options are listed.

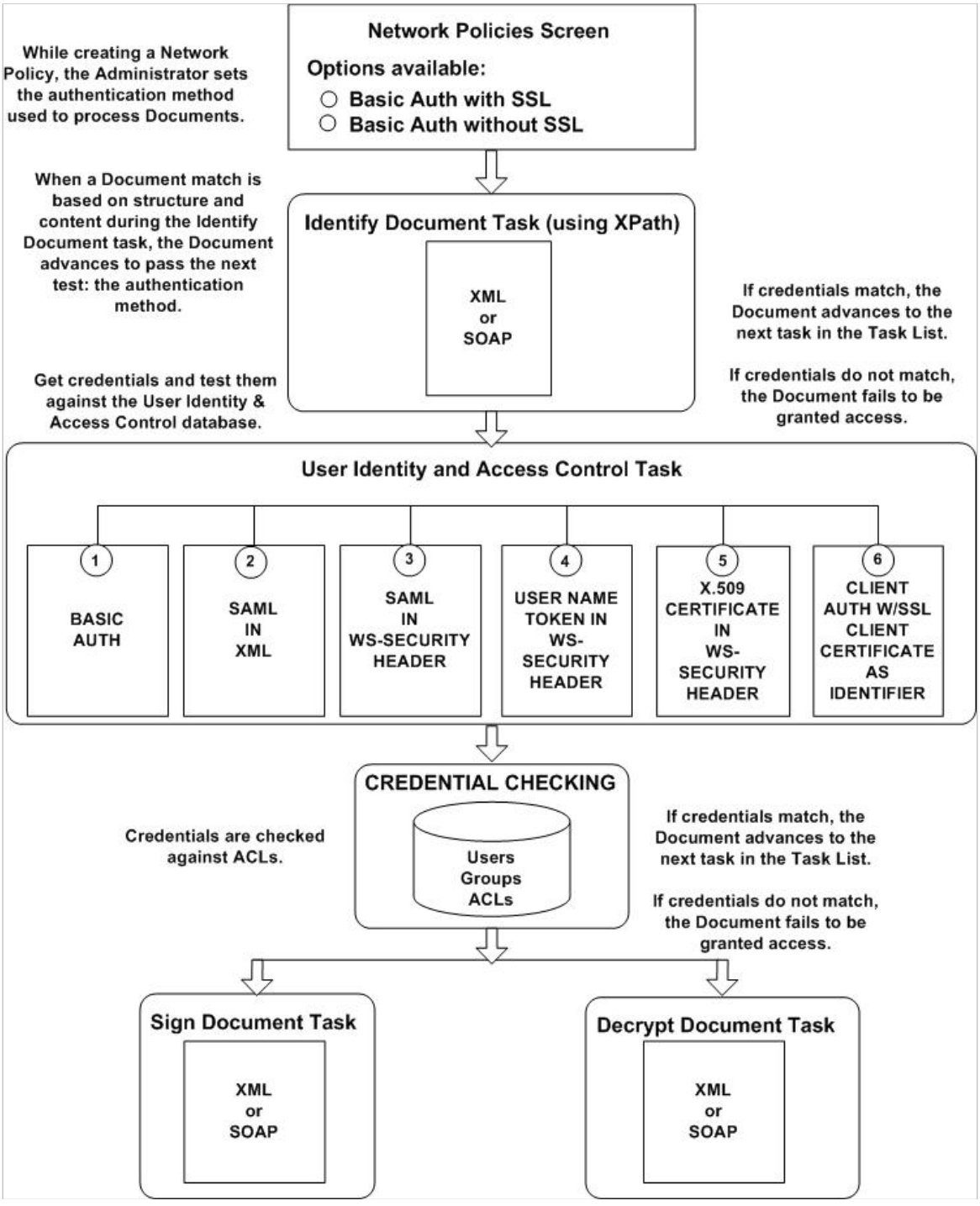


Figure 1: Credential Binding Options Interact with Network Policies.

Transport-centric and Document-centric Mechanisms

The following table displays some of the transport-centric and document-centric mechanisms supported by the system. Not all access control options are listed.

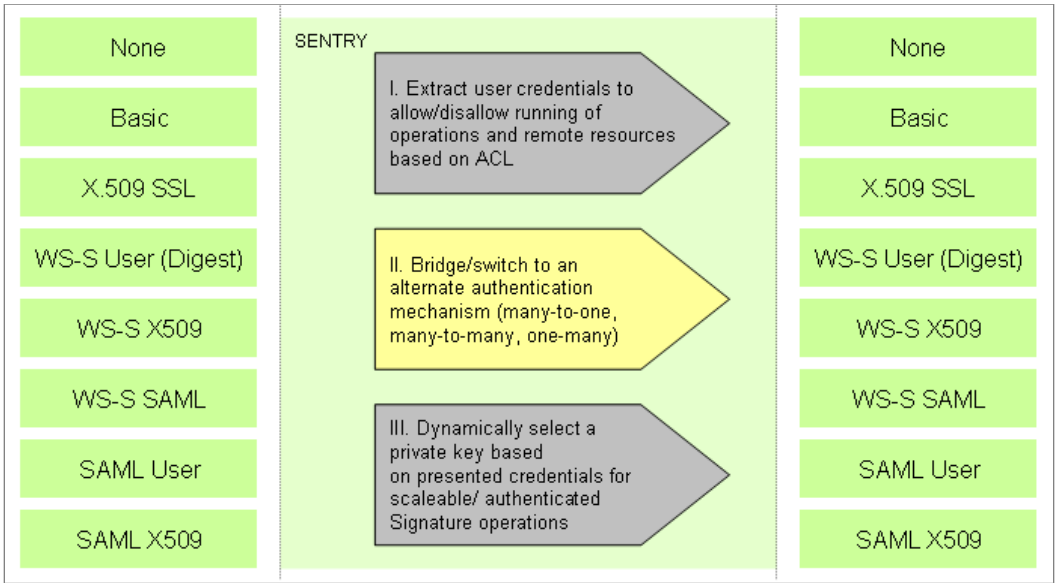


Figure 2: Transport-Centric and Document-Centric Mechanisms Supported on the Product.

WS-Security Headers

A WS-Security Header is a mechanism for conveying security information with and about a SOAP message. This header is, by design, extensible to support many types of security information. For security tokens based on XML, the extensibility of the WS-Security header allows for these security tokens to be directly inserted into the header.

WS-Security Header with User Name and Password

A common method for passing a client's credentials is to use a username and password.

With WS-Security, passing user name / password is handled with the User Name Token, as in the following example:

```
<wsse:UsernameToken wsu:Id="feba6d12fdc1c7d8e2d12f82319b2d860795738c"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
    wssecurity-utility-1.0.xsd">
     <wsse:Username>markcross</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss
            2004/01/oasis-200401-wss-username-token-profile-1.0
            #PasswordText">markcross
         </wsse:Password>
           <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss
              /2004/01/oasis-200401-wss-soap-message-security-
              1.0#Base64Binary">taHydP2Y3mMWG7kzjxtfxQ==
           </wsse:Nonce>
              <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss
                 /2004/01/oasis-200401-wss-wssecurity-utility-
                 1.0.xsd">2005-02-09T09:54:34Z
              </wsu:Created>
```

```
</wsse:UsernameToken>
```

Username and Password are strings that contain extra attributes, as needed. The Password element contains the Type attribute that indicates how the password is being passed around. The Password may be passed as plain text or in digest format.

The following displays the User Name Token in a SOAP message with a Password being passed as plain text:

```
<wsse:UsernameToken>
    <wsse:Username>Chris</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/
           2004/01/oasis-200401-wss-username-token-profile-
           1.0#PasswordText">password</wsse:Password>
</wsse:UsernameToken>
```

The following displays the User Name Token in a SOAP message with a Password being passed as a digest hash:

```
<wsse:UsernameToken>
    <wsse:Username>Chris</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
            oasis-200401-wss-username-token-profile-1.0#Password
            Digest">G9310wl8fPRPvqvOZGSFJe0YQxs=</wsse:Password>
               <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss
                  /2004/01/oasis-200401-wss-soap-message-security-1.0#
                 Base64Binary">wrDrkMbqa2dDQlqybfX9cw==</wsse:Nonce>
                     <wsu:Created xmlns:wsu="http://docs.oasis-
                        open.org/wss/2004/01/oasis-200401-wss-
                        wssecurity-utility-1.0.xsd">2005-02-
                        07T20:51:24Z</wsu:Created>
</wsse:UsernameToken>
```

WS-Security Header with X.509 Certificate

Clients may be authenticated with an X.509 Certificate, indicating exactly who the user is.

With a WS-Security Header combined with an X.509 Certificate, the originator may sign the Document with a Private key. The X.509 Certificate is passed as base64 encoded data which can be used to map to a user, or user directly to authenticate based on the X509 Path Validation.

With WS-Security, passing an X.509 Certificate is handled with the BinarySecurityToken token, as in the following example:

```
<wsse:BinarySecurityToken EncodingType="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
  1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/
  2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
  wsu:Id="cb2f90ea97b477de1647a04c3c1408ed01ed173b"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
  wssecurity-utility-1.0.xsd">MIIBLjCB2aADAgECAgEAMA0GCSqGSIb3D
  QEBBQUAMBUxEzARBgNVBAMTCmphY2trYW50b3MwHhcNMDUwMjA5MDkzNzUyWhcNMDYwMj
  A5MDkzNzUyWjAVMRMwEQYDVQQDEwpqYWNra2FudG9zMFowDQYJKoZIhvcNAQEBBQ
  ADSQAwRgJBAKPdLw2xBD1OV0vNOVXMZvTJ8yckzzuSehSU+PFDJldOzAQJiagm
  Dc9i59Lq7Tg7qPU+m2epAJq++g6qMbH4kR0CARGjFjAUMBIGA1UdEwEB/wQIM
  AYBAf8CAQAwDQYJKoZIhvcNAQEFBQADQQCdq25dX8+dv1WUC2zgWfQtY5awFMZ0Gy
  1w56ktrQPka8xE3BTglmfsug2O3QWdRm/1Umdz53urAb+fi/xyMUg6
</wsse:BinarySecurityToken>
```

WS-Security Header with SAML

SAML Assertions are attached to SOAP documents using WS-Security by placing the Assertion elements inside the ***<wsse:Security>*** header. These assertions may be SAML Assertion Email tokens or SAML Assertion Distinguished Name tokens. The following SOAP message contains a SAML Assertion with Email in a WS-Security header:

```
<S:Envelope xmlns:S=”…”>
    <S:Header>
      <wsse:Security xmlns:wsse=”…”>
          <saml:Assertion
             MajorVersion=”1”
             MinorVersion=”1”
             AssertionID="22fceef94df83c03b86288244ccfc5fbb94ae9b9"
             Issuer=””
             IssueInstant="2005-02-07T20:55:54.265Z”
             xmlns:saml=”urn:oasis:names:tc:SAML:1.0:assertion”>
          </saml:Assertion
      </wsse:Security>
    </S:Header>
  <S: Body>
...
  </S:Body>
<S/:Envelope>
```

Note: With the WS-Security Header task, the Administrator is creating a series of rules about authentication that will apply to an XML document coming into the product.

For more information on WS-Security Header tasks, refer to the *Forum Systems Sentry™ Version 9.1 Tasks Management Guide*.

SAML Assertions

SAML provides Single Sign-on, which allows users to be authenticated between security domains.

If, for example, there are three servers; A, B, and C, and each is in a different security domain:

- Server A trusts Server B, who trusts Server C.
- Server C authenticates you and generates an assertion, and then Server C passes your document and the assertion to Server B.
- Server B receives the document and the assertion, and because Server B trusts Server C, and the assertion is valid, Server B does not require that you personally authenticate to it. Server B could either:
    - send the assertion and document on to Server A (if A trusted C), or
    - generate another assertion indicating that it has authenticated you.
- Server A receives the assertion and document, and the transaction is complete.

Because SAML assertions travel with a document, the user is authenticated once while the SAML assertions are passed to each hop.

SAML assertions also make a statement about a subject (user or service). All SAML assertions include the following common data:

- Issuer and issuance timestamp
- Assertion ID
- Subject
- Name plus security domain.
    - Optional subject confirmation (e.g., public key).
    - Conditions (under which this SAML assertion is valid).
- Advice (concerning how the SAML assertion was made).

SAML specifies three components: 1) assertions, 2) protocol, and 3) binding. Each of these components is discussed next.

There are three types of SAML assertions:

1. Authentication
   A ***SAML Authentication Assertion*** is a statement that asserts that the user has been authenticated by the sender/creator of the assertion. SAML Authentication Assertions also include an issuing Authentication Authority that asserts that subject S was authenticated by method M at time T. Credentials are then requested, checked and passed to the product.
2. Attribute
   SAML Attribute assertions contains specific information about the user. A **SAML Attribute Assertion** includes an issuing Attribute Authority asserts that subject S is associated with Attributes X, Y, etc. with values a, b, etc. Example: Subject “John Doe” is associated with Attribute “Department” with value “Developer”.
3. Authorization
   SAML Authorization assertions identify what the user is authorized to do. A **SAML Authorization Assertion**includes an issuing Authorization Authority decision to grant a request by subject S for Action A on Resource R (a file, an application, a Web service, etc.) given Evidence E.

SAML Protocols
Protocol defines how SAML asks for and receives assertions. SAML works with multiple protocols including:

- Hypertext Transfer Protocol (); currently supported by Sentry.
- Simple Mail Transfer Protocol ()
- File Transfer Protocol ()

SAML Bindings
Binding defines how SAML request-response message exchanges are mapped to Simple Object Access Protocol () exchanges.

SAML Assertion Task
With the SAML Assertions task, the Administrator is generating a series of rules, called a SAML Assertion, onto an XML document coming into the product. These assertions may be SAML Assertion Email tokens or SAML Assertion X.509 Distinguished Name tokens.

Note: For more information on SAML Assertion tasks, refer to the *Forum Systems Sentry™ Version 9.1 Tasks Management Guide*.

Password Authentication (HTTP Authentication) Overview
The Password Authentication options available with HTTP(S) Listener Policies are: Basic Authentication, Digest Authentication, Cookie Authentication, and Form Post Authentication. These are all considered "HTTP protocol level' authentication mechanisms. These options can be required or optional, and the user can specify a Password Authentication Realm.

Sentry provides the following three protocol authentication types when authenticating the user. These are listed from least security to most secure:

- Password Authentication without SSL
- Password Authentication over SSL
- Password Authentication over SSL with Client SSL Certificates

These three protocol authentication types are defined next:

Password Authentication without SSL
The client provides the credentials over an HTTP connection which is not secured (encrypted) with SSL..

Password Authentication over SSL
The SSL protocol encrypts the logon session. SSL Initiation and SSL Termination policies created in the product provide another layer of authentication.

Password Authentication with SSL with Client Certificates
The user connects with SSL. As part of that SSL negotiation, their public key certificate is transmitted to the server. The credentials are then transmitted over a secured connection. With HTTP Basic Authentication enabled, upon successful SSL connection, the user is then prompted for their Username and Password. Upon successful authentication, the request is completed.

Password Authentication at the WSDL Policy Level
The Password Authentication options are also available with WSDL Policies, with the administrator having the option to either use the Password Authentication options "From Listener Policy" or set them at the WSDL Policy level.

Basic Authentication Type

Basic Authentication is one of the more commonly used Password Authentication options with HTTP(S) listener policies. Basic Authentication allows incoming clients to authenticate themselves to destination servers or resources. HTTP Basic Authentication, commonly called Basic Auth, is the authentication scheme included in the HTTP/1.1 standard. Basic Authentication allows clients to pass their Username/Password to another server. By itself, Basic Authentication is not considered secure, unless used in tandem with a transport level security mechanism like, SSL.

With *Basic Authentication*, the client passes a Username and Password to the server as a single base64-encoded block, so the Username and Password are transmitted across the wire unencrypted.

Digest Authentication Type

Digest Authentication (also known as Digest access authentication or just Digest Auth) is one of the agreed methods a web server can use to negotiate credentials with a user's web browser. It uses encryption to send the password over the network which is safer than the Basic Authentication which sends base64-encoded plain text. While slightly more secure than Basic Auth, use of SSL is still suggested for additional security.

Cookie Authentication Type

Cookie Auth uses one of the known cookie types to provide single-sign on authentication based on a previously authenticated user causing the cookie to be generated and returned to the client. Supported cookie tokens include SMSESSION (CA Siteminder), ObSSOCookie (Oracle Access Manager), and FSSESSION cookie (Forum Sentry). Forum Sentry cookies provide localized caching and session management options for Federation and single-sign-on scenarios where any back-end identity system can be used to create session based authentication federation.

Form Post Authentication Type

Form Post Authentication is the setting used to extract credentials from an HTTP web form POST request. This is often used in Portal type use cases where an HTML page is presented to the client requesting credentials. The POST request is submitted to Sentry as a Form Post authentication request. Settings for this authentication type include Username and Password field names where the credentials are to be extracted from the incoming HTML.

Access Control and Credential Binding

Credential binding is a method of configuring the product to dynamically apply actions based on the identity of a user. This allows a single task group to have applicability across defined User policies which map to an Access Control List (ACL). The following graphic displays the interactions of credential binding and dynamic signing. *

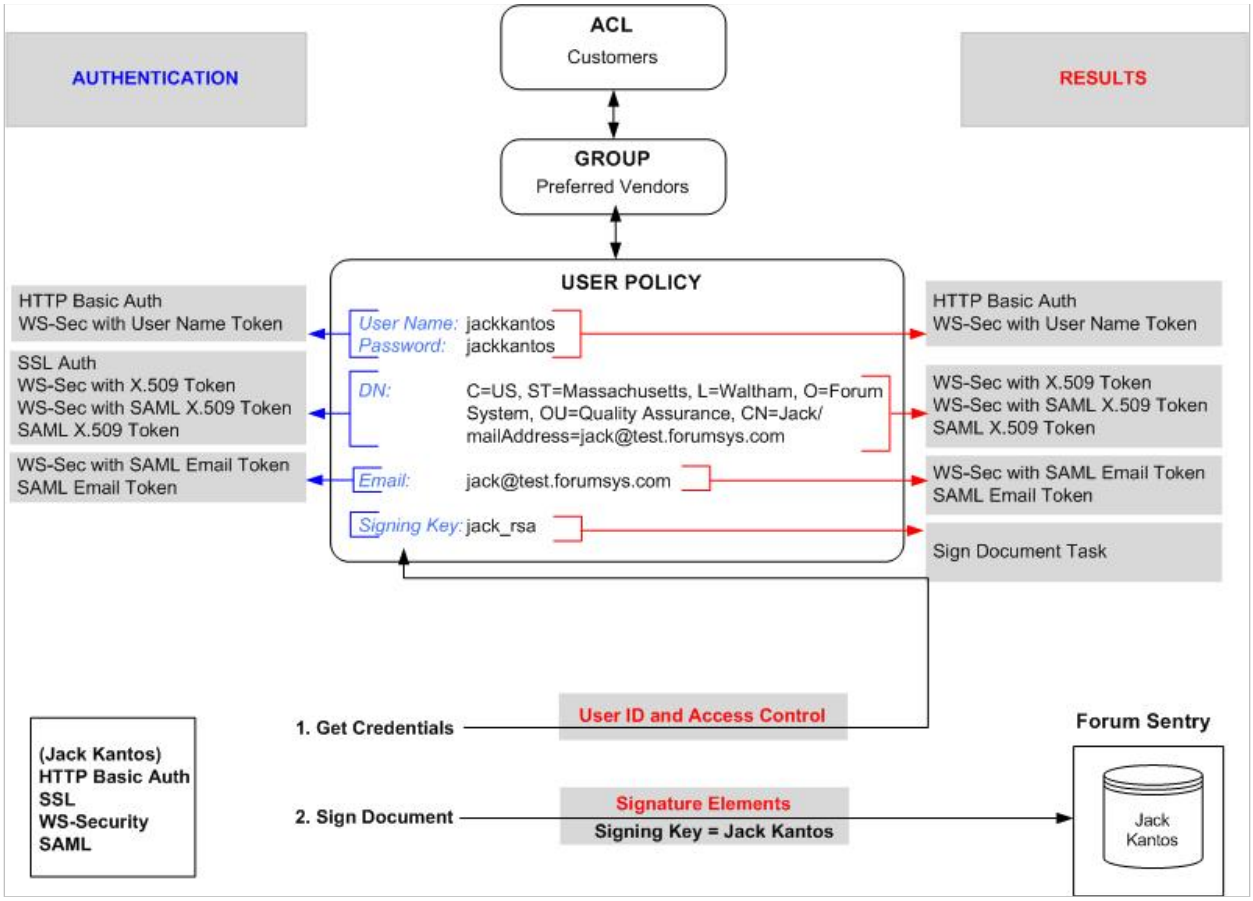


Figure 3: Dynamic Signing with Credential Binding.

Note: For more information on ACLs, refer to the ACL Policies section.

Access control and authorization is unique in that the product supports:

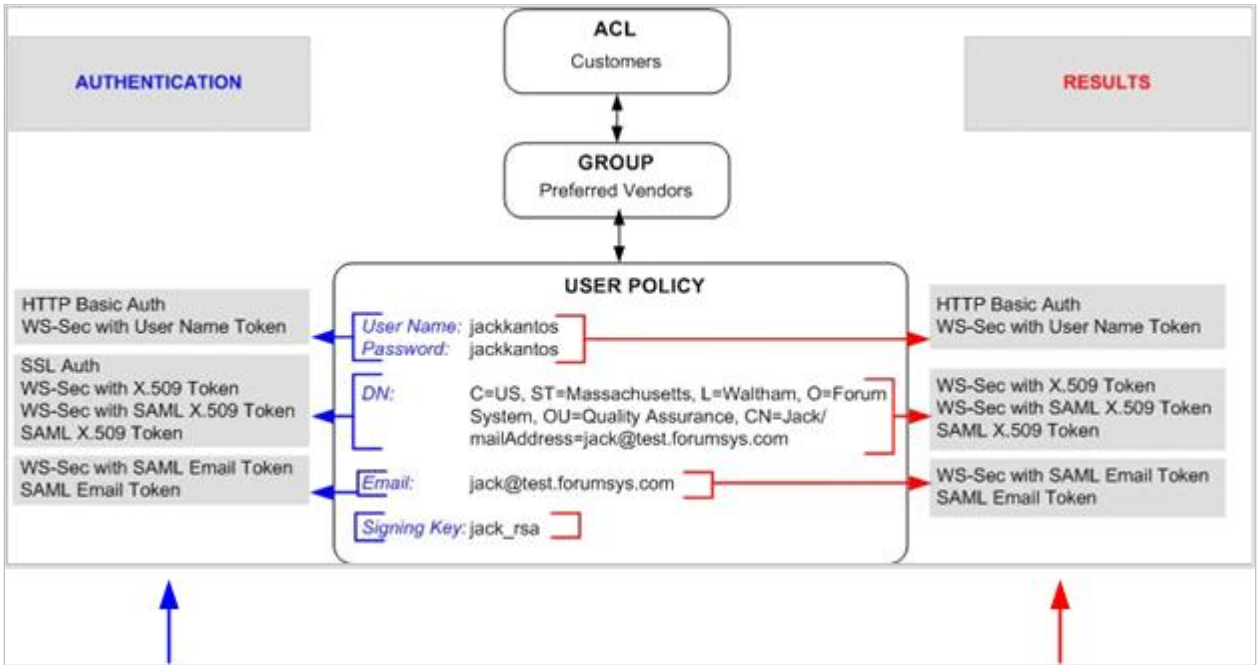
- Transport-centric mechanisms – HTTP Basic Auth., HTTP Digest Auth., HTTP Cookies, Form post Auth, and SSL Client Certificates.
- Document-centric mechanisms – SAML and WS-Security header credentials, such as Username Token, X.509 Binary Token, X.509 Distinguished Name Token, and Email Token.

The major advantages of access control and authentication are:

1. During the User Identity & Access Control task, the product extracts user credentials to allow/deny running of operations and remote resources based on an ACL in which a user shared membership.
2. The User Identity & Access Control task acts as a bridge/switch to alternate authentication mechanisms to A) Many-to-one; B) Many-to-many; and C) One-to-many.
3. During the Signature operation, the product uses a Private Key based on policy for scalable / authenticated Signature operations.

Advantages of Access Control and Authentication

The following graphic displays authentication headers supported on the product:



Adding authentication headers (the left side of the above figure) map to a User Policy. Users belong to Groups that have membership in ACLs.

Through ACLs, users are identified and granted access.

Generating authentication headers (the right side of the above figure) map to a User Policy. Users belong to Groups that have membership in ACLs.

Through ACLs, users are identified and granted access.

Figure 4: Building Authentication Headers Based on Authentication Method.

Path Validation and Signer Groups

The following graphic displays four scenarios that reveal path validation with Signer Groups:

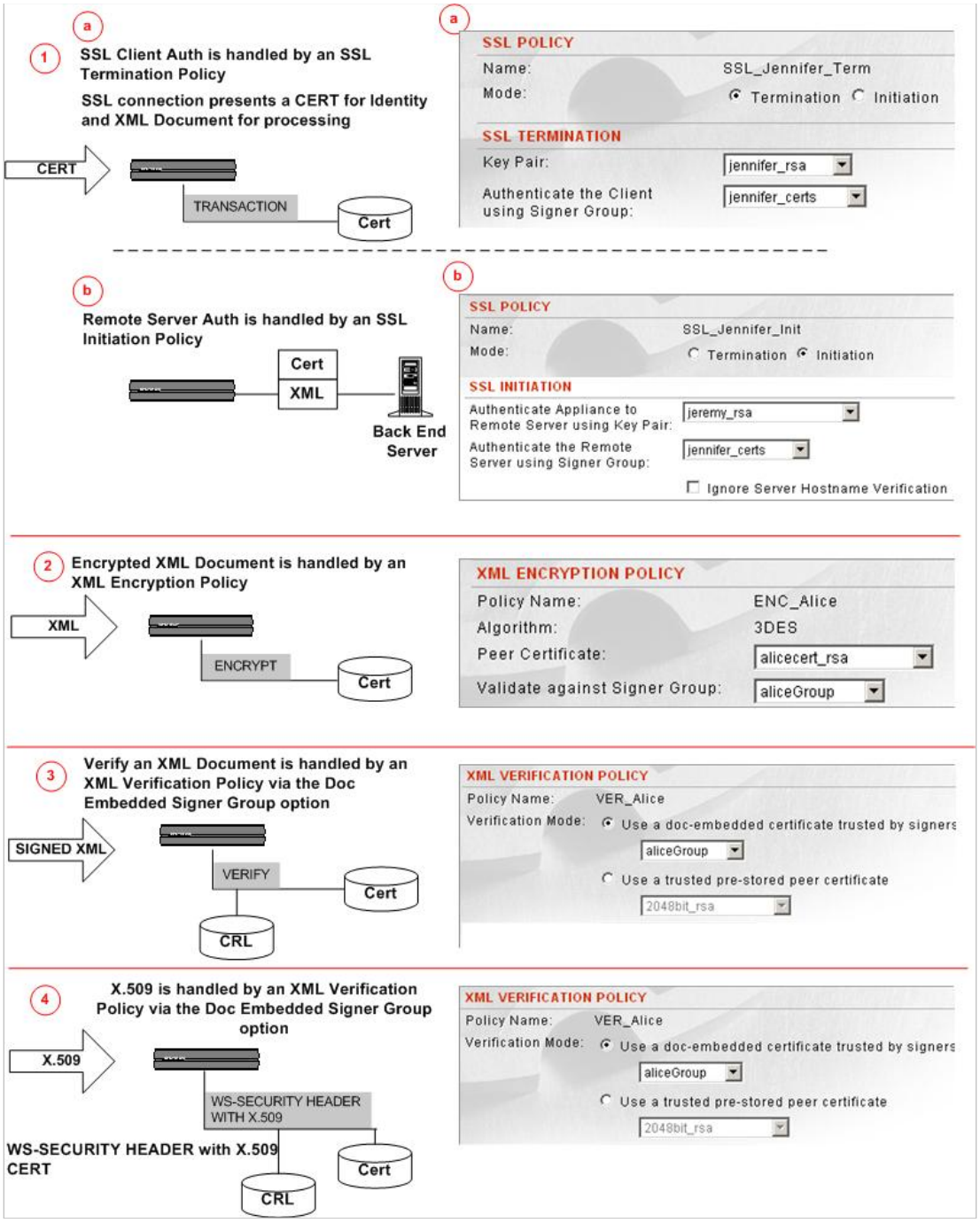


Figure 5: Four Scenarios for Using Signer Groups.

Note: For more information on Signer Groups, refer to the *Forum Systems Sentry™ Version 9.1 Security Policies and PKI Guide*.

LDAP AND ACTIVE DIRECTORY POLICIES

The LDAP screen provides initial configuration to accommodate LDAP v3 standard schemas or Microsoft Active Directory schemas. It also allows custom modifications to map to other LDAP schemas. Administrators may enable, disable, save or delete LDAP server configuration settings; update configuration settings and test validate connectivity. LDAP Policies can be used for either LDAP or Active Directory user containers and subsequent sections describing LDAP apply to both LDAP v3 compliant identity stores as well as MS Active Directory identity stores.

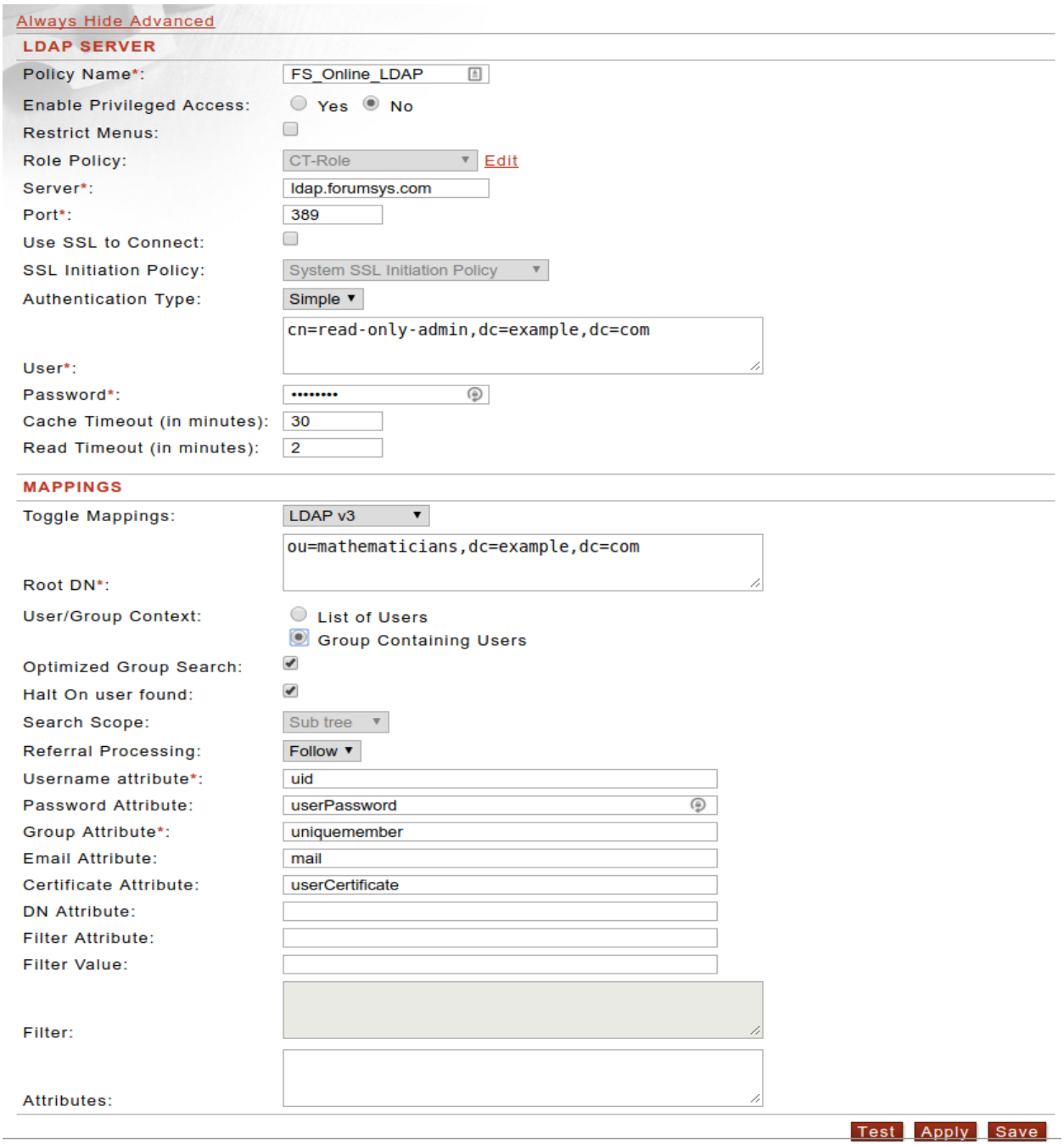
LDAP Screen Terms

The terms and definitions that are listed on the LDAP screen include the following:

FIELD NAME	DEFINITION
Policy Name	The identifier for this LDAP policy.
Status	• Green status light = enabled policy. • Yellow status light = a required functional element of this policy is disabled. • Red status light = disabled policy.
LDAP Server	The IP address of the LDAP Server on this LDAP policy.

LDAP Policy Views

When selecting an existing LDAP Policy link from the LDAP POLICIES screen, the LDAP POLICY screen appears. This screen provides two views. By selecting the **Always Show Advanced** or **Always Hide Advanced** links. The Advanced View allows users to enter additional values.



LDAP Policy Details Screen Terms

When configuring your LDAP server from the LDAP details screen, please consider the following:

FIELD NAME	DEFINITION
LDAP SERVER	
Policy Name	The identifier for this LDAP policy.

Enable Privileged Access	• With **Yes** selected, the user has access to the WebAdmin as a super user. • With **No** selected, the user still have access to the WebAdmin with all Domain privileges set for the Group which this user is a member of.
Server	The host name or IP address of your LDAP server.
Port	The port number for your LDAP server. The default port is 389. The SunONE default port is 389, the Oracle default port is 4032, and the LDAP default port is 636. SSL is used to connect to the LDAP server.
Use SSL to Connect	• With Use SSL to connect checked, SSL is used to connect to the LDAP server. SSL connections are initialized using the SSL Initiation policy selected below. • With Use SSL to connect unchecked, your LDAP server connects to the system via non-SSL connections.
SSL Initiation Policy	SSL Initiation Policy that will be used to connect to the LDAP server
Authentication Type	There are three authenticating mechanisms: None, Simple and SASL. • The None option means that the system does not authenticate to the LDAP server. No username or password are required with this option. • The Simple option means that the system binds to LDAP using a username and password. • Simple Authentication Security Layer (SASL) is a method for adding authentication support to connection-based protocols. The SASL option means that the system authenticates to the LDAP server using the SASL authentication mechanism. **Note:** The Forum Systems implementation of SASL uses the Digest authentication mechanism.
User	The username for requesting data from the LDAP server. With Authentication type of None, no username is required.
Password	The password for the user requesting data from the LDAP server. With Authentication type of None, no Password is required.
Cache Timeout (in minutes)	The cache timeout in minutes is the amount of time the system caches user credentials on the system. This may be from 0 to 120 minutes.
Read Timeout (in minutes)	The read timeout in minutes is the amount of time the system will wait for data after connecting to the LDAP server. This may be from 0 to 120 minutes.

FIELD NAME	DEFINITION
MAPPINGS	
Toggle Mappings	Toggles the values for username and group attributes in the advance section. LDAP and Active Directory use different attributes to refer to these fields and the selected toggle (LDAP v3 or Active Directory) populates the values to correct defaults for each case. These defaults can then be edited further if necessary to connect to the specific target identity container.
Root DN	Root DN is the point at which you want to retrieve LDAP users and groups from your LDAP tree. You must specify in the Root DN field an exact context that is the valid path in your LDAP directory to start the search.
User / Group Context	The “List of users” option interprets the root DN specified as the base DN to build for the incoming user credentials. With the provided user, the full DN is constructed containing the LDAP Root DN, the user attribute specified, and the user information provided. In this case where the provided information includes the user name, the format of the DN object to be queried would be determined by: <uid>=username, <root dn> The “Group containing users” option provides the means to specify a context of an LDAP Group object. This object is then dynamically used when user credentials are provided to match with a user record found within the group object's uniquemember attribute.
Optimized Group Search	Option available only when using the “Group containing users” option. Allows for specifying a group attribute to optimize the group search.
Halt On user found	If this option is enabled, then when processing an ACL containing this LDAP policy, if the user is found under this LDAP policy and the password is incorrect then processing of the ACL stops.
Search Scope	Option available only when using the “List of users” option. Allows for specifying a search scope level of Sub tree, one level, or object.
Referral Processing	Determine how Sentry will process LDAP referral entries found during queries. Sentry can Follow the referral, ignore it or Fail the query when a referral is found in response to a query.
Username Attribute	In the LDAP tree structure, every node contains attributes which are defined by a schema. The Username attribute is the attribute used to identify the username value for the user object.
Password Attribute	The Password attribute is the attribute used to identify and verify the user password.
Group Attribute	The Group attribute is the attribute used to identify an LDAP group object.
Email Attribute	The Email attribute is the attribute used to identify the email address value of the user object.
Certificate Attribute	The Certificate attribute is used to identify the Certificate.
DN Attribute	The DN attribute assumes entryDN (i.e., the actual DN context of the object in LDAP), or this value can point to a different context by entering an attribute for this value (for matching a Certificate published under a different context).

Filter Attribute	Using the Filter Attribute and Filter Value below the LDAP entries returned by a query to select a smaller subs-set of records (user objects). Example: • Filter Attribute: employeeType • Filter Value: Engineer
Filter Value	The Filter Value represents the value set for the Filter Attribute being constrained.
Filter	The filter represents and a string representation of an LDAP search filter as defined in RFC 2254. The following interpretations of attr and value mentioned in the RFC are used: Where attr is the attribute's identifier value is the string representation of the attribute's value and the translation of this string representation into the attribute's value is directory-specific. Any non-ASCII characters in the filter string should be represented by the appropriate Unicode characters, and not encoded as UTF-8 octets. Alternately, the "backslash-hexcode" notation described in RFC 2254 may be used. RFC 2254 defines certain operators for the filter, including substring matches, equality, approximate match, greater than and less than. These operators are mapped to operators with corresponding semantics in the underlying directory. With the “equals” operator, for example, suppose the directory has a matching rule defining "equality" of the attributes in the filter. This rule would be used for checking equality of the attributes specified in the filter with the attributes of objects in the directory. Similarly, if the directory has a matching rule for ordering, this rule would be used for making "greater than" and "less than" comparisons. Not all of the operators defined in RFC 2254 are applicable to all attributes. When an operator is not applicable, it will cause an error. Examples: (cn=Babs Jensen) (!(cn=Tim Howes)) (&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*))) (o=univ*of*mich*) For more information, review RFC 2254 - The String Representation of LDAP Search Filters at .
Attributes	Additional attributes to be retrieved and stored on authentication of the user.

* If you require interoperability with non-standard schemas, you can edit entries on the user object classes and group object classes. The LDAP screen appears with standard object names and attributes that can be modified to fit your schema.

How LDAP Policies Access Information

With LDAP policies, no information is stored on the system; instead, all information is retrieved dynamically from LDAP.

Access Control and LDAP Group Privileges

Access privileges may be set for an LDAP policy from the User ACLs screen, which can be used to grant Execute privileges. The User ACL details screen displays the privileges enabled for the LDAP policies.

The LDAP policy itself represents a population of users. The LDAP policy will appear in the User ACLs screen just as a standard group will appear, and Administrators can associate Read, Write (for Administration) and Execute (for run-time processing) privileges to the LDAP policy.

Working with the List of Users Option

The LDAP "List of users" option interprets the root DN specified as the base DN to build for the incoming user credentials. With the provided user and the specified user attribute, the DN is searched at scope of one level for a matching user attribute, or all levels if the scope is extended. Membership is determined by the DN context (location) of the user object being found under the specified DN with the specified scope.

The Search Scope feature can be used to optimize the LDAP query when using the List of Users option. The options are sub tree, one level, and object.

Working with the Group Containing Users Option

The LDAP policy "Group containing users" option provides the means to specify a context of an LDAP Group object. This object is then used when user credentials are provided to match with a user record found within the specified group object' member attribute. Membership is determined based on user object referenced in a group member list. The member list attribute is configurable on the LDAP properties.

The Optimized Group Search feature can be used to optimize the LDAP query when the Group Containing Users option is enabled. This works in conjunction with the Group attribute.

Authentication and Authorization for LDAP Users

Sentry will authenticate LDAP users stored in LDAP v3 compliant systems such as SunONE, iPlanet, RSA Keon, MS Active Directory, Oracle Directory or any other LDAP v3-compliant directory.

In the cases of HTTP Basic Authentication and WS-Header Username authentication, a password is supplied with the credentials. For these cases, the password can be validated by attempting to bind directly to the LDAP user object with the supplied username and password. If any other authentication scheme is being used where the user password is not available, the LDAP policy credentials are required to access the LDAP user object.

Sentry determines how to bind to LDAP per the User ACL policies configured with LDAP policies as members.

LDAP policies and ACL policies work in tandem to provide user authentication and authorization for LDAP users. LDAP policies validate user membership in the LDAP Policy or "Group" (Authenticate). ACLs are configured to allow or disallow Groups (Authorize).

LDAP Policies Examples

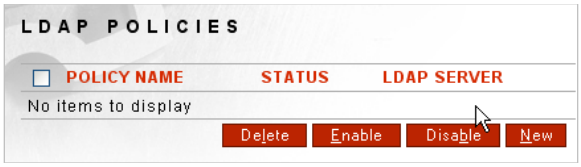
Examples for LDAP policies include:

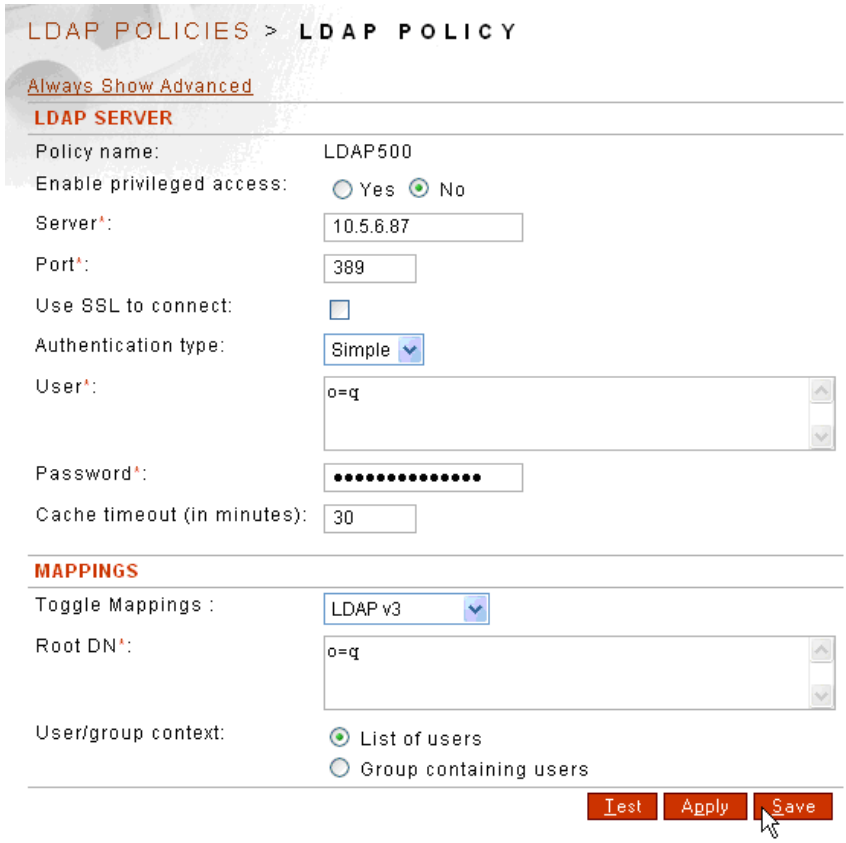
- Add an LDAP Policy with List of Users Option.
- Assign Privileges for Run-time Access.
- Test an LDAP Policy.

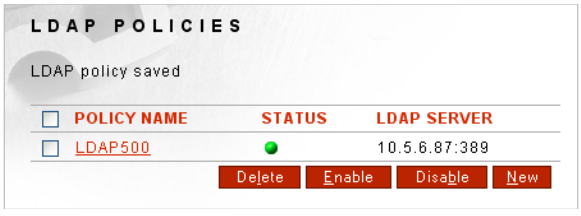
Add an LDAP Policy with List of Users Option

Users may create an LDAP policy with LDAP server configuration settings. This action only saves the values entered on this screen. Credential verification is performed dynamically at execution time, no user information is stored locally.

Before creating an LDAP policy with the List of Users option, create a User ACL which will allow access to this list of users on the LDAP policy.







- Navigate to the **LDAP** screen.
- On the LDAP POLICIES screen, click **New** and the LDAP POLICY screen appears.
- In the Policy Name field, enter a **name** for this LDAP policy.
- Aligned with Enable privileged access, select the **No** radio button. This setting determines whether the LDAP group is to be used for runtime or administrative users. If the policy is to be used for administrative purposes, this setting could be set to Yes to provide superuser access privileges to administrators who belong to this policy.
- In the Server field, enter the LDAP Server **host name** or **IP address**.
- In the Port field, either retain the default port of 389 or overwrite this port number with the **port number** of your LDAP server.

Note: LDAP Server port may be an integer between 1 and 65535.

- Check the **Use SSL to connect** checkbox to connect using SSL, or skip this field if not connecting through SSL.
- If you are connecting through SSL you will need to select an existing **SSL Initiation** policy to use when connecting to the LDAP server.
- From the Authentication Type drop down list, select **None**, **Simple** or **SASL**.
- In the User text box, enter the **LDAP context** for the user who is presenting authentication credentials to the LDAP serve.
- In the Password field, enter the **user's password**.
- Accept the default (30) in the Cache timeout in minutes' value.
- From the Toggle Mappings drop down list, select the **LDAP v3** or **Active Directory** option.
- In the Root DN field, enter the **LDAP context** from which you want groups and users fetched on the LDAP server.

Note: The Root DN may be from 0 to 255 alpha characters, may include spaces, period and at least one equal sign. Multiple groupings must be separated by commas. Example: Root DN = ou=People, o=qa-lab.forumsys.com.

- Aligned with User / Group context, select the **List of users** radio button.
- Select **Save** and the LDAP POLICIES screen refreshes, or select **Apply** to save this LDAP policy and remain on this screen.

Assign Privileges for Run-time Access

The LDAP group will be prefixed with LDAP- and the name provided for the policy. This name subsequently appear in the list of available Groups on the ACL DETAILS screen allowing you to associate Execute privileges to this LDAP group which in turn make the LDAP Group a member of the selected ACL.

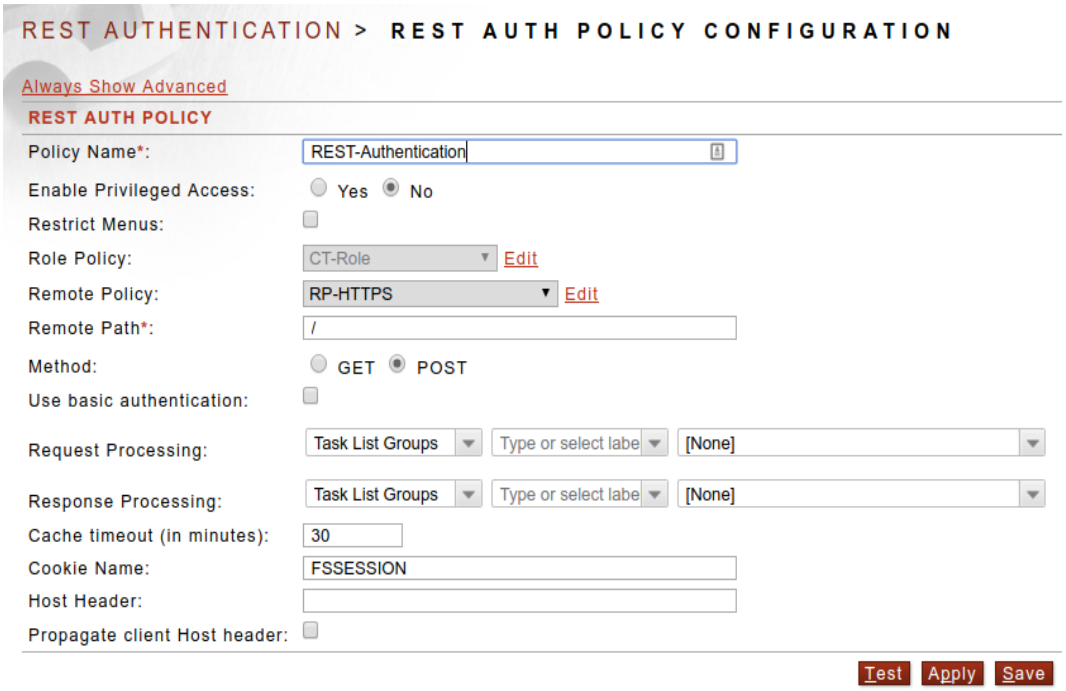
Test an LDAP Policy

Administrators may test the LDAP policy by confirming that connectivity is successful with the credentials presented.

- Navigate to the **LDAP** screen and select an **LDAP policy name** link.
- On the LDAP POLICY screen, select **Test** and the LDAP POLICY screen refreshes with the "Successfully authenticated with LDAP server" message at the top of the screen.

REST AUTHENTICATION

REST Authentication policies allow Sentry to act as a client to a REST service which will indicate to Sentry if a particular user is valid based on HTTP request.



REST Authentication Policy Configuration

When configuring your REST Authentication policy, please consider the following:

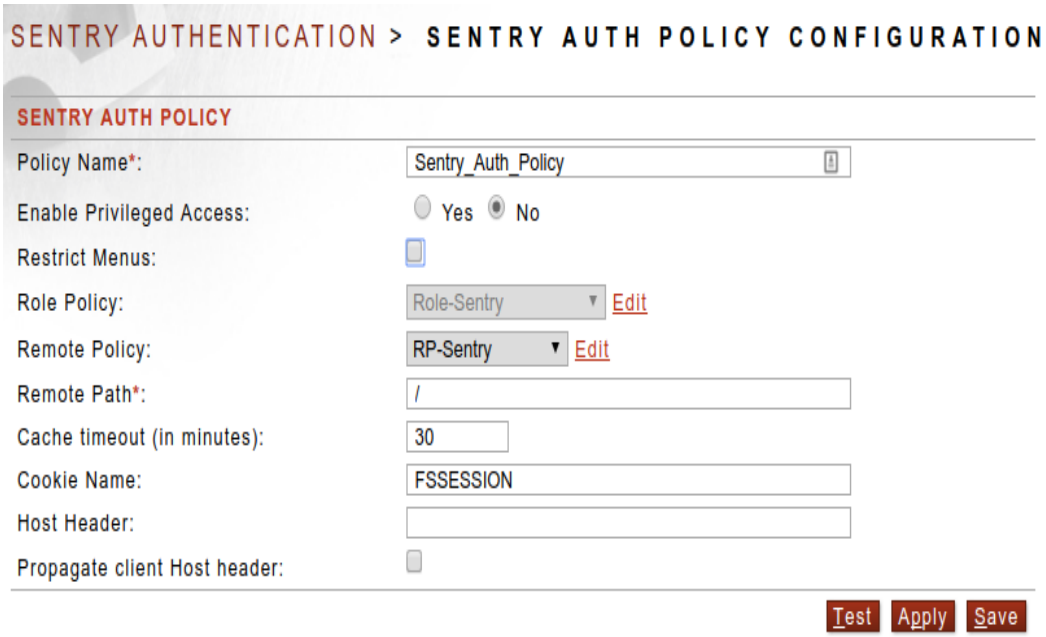
FIELD NAME	DEFINITION
REST AUTH POLICY	
Policy name	The identifier for this REST Authentication policy.
Enable privileged access	• With **Yes** selected, the user has access to the WebAdmin as a super user. • With **No** selected, the user still have access to the WebAdmin with all Domain privileges set for the Group which this user is a member of.
Restrict Menus	If using this policy for WebAdmin users, you can restrict menus for the users.
Role Policy	If using this policy for WebAdmin users, you can specify a defined Role for the users.
Remote Policy	You will need to identify a Remote Policy which defines the remote IP or Host and the port for the remote REST service this authentication policy will use..
Remote Path	The remote path to use when connecting to the Remote Policy. The remote policy IP or Host, the port, and the remote path make up the full URI for the remote REST service this authentication policy will use.
Method	Determines if the request sent to the REST Authentication server will use the GET or POST HTTP method.

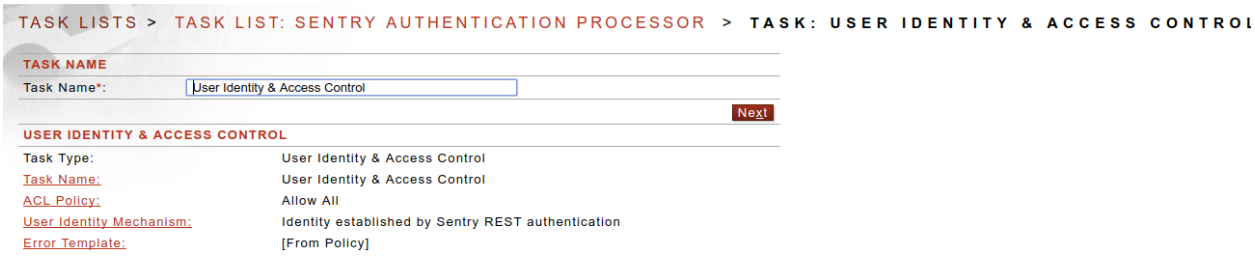
Use basic authentication	Enables the sending of HTTP Basic Authentication credentials provided by the client to the REST Authentication server.
Request Processing	Sets up processing of the request so it can be structured properly for the REST authentication server.
Response Processing	Sets up processing of the REST Authentication server response to perform actions like converting entries in the response into user attributes.
Cache Timeout	How long Sentry will cache the results of the REST call to validate credentials. The cache value is in minutes, the default value is 30.
Cookie Name	The name of the cookie set by Sentry and returned to the client.
Host Header	The optional Host Header set by Sentry.
Propagate client Host header	This check box allows the Host header from the inbound request to be sent to the REST authentication remote server.
User Name Parameter	Identity attribute into which the username will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
Password Parameter	Identity attribute into which the user's password will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
Sha1 Parameter	Identity attribute into which the SHA1 hashed password will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
Cookie Name Parameter	Identity attribute into which the cookie name will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
Cookie Value Paramter	Identity attribute into which the cookie value will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
X.509 Certificate Parameter	Identity attribute into which the user's X.509 certificate will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
DN Parameter	Identity attribute into which the user's DN will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.
Email Parameter	Identity attribute into which the user's email address will be stored for addition to the user session. This is done via mapping tasks in the request or response processing.

REST Authentication polices are similar to LDAP or SiteMinder Policies in that they will show up as Groups under the User ACLs. These policies can be used for either run time users or Design time users (WebAdmin access).

SENTRY AUTHENTICATION

Sentry Authentication policies allow Sentry to act as a client to a Sentry with a specially setup REST policy. The REST Policy on the other Sentry that will authenticate the request must have a task list which contains a "User Identity & Access Control" task that uses the desired ACL and is set to use the "Identity established by Sentry REST authentication" user identity mechanism





Sentry Authentication Policy Configuration

When configuring your Sentry Authentication policy, please consider the following:

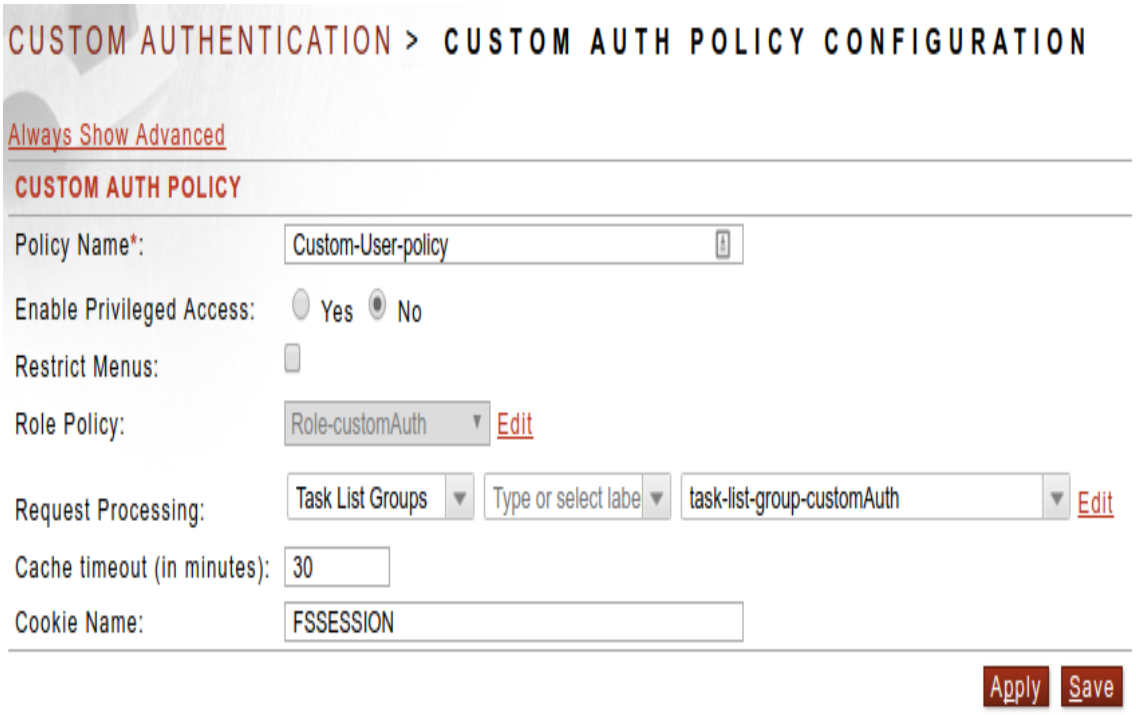
FIELD NAME	DEFINITION
SENTRY AUTH POLICY	
Policy name	The identifier for this Sentry Authentication policy.
Enable privileged access	• With **Yes** selected, the user has access to the WebAdmin as a super user. • With **No** selected, the user still have access to the WebAdmin with all Domain privileges set for the Group which this user is a member of.
Restrict Menus	If using this policy for WebAdmin users, you can restrict menus for the users.
Role Policy	If using this policy for WebAdmin users, you can specify a defined Role for the

	users.
Remote Policy	You will need to identify a Remote Policy which defines the remote IP or Host and the port for the remote Sentry this authentication policy will use..
Remote Path	The remote path to use when connecting to the Remote Policy. The remote policy IP or Host, the port, and the remote path make up the full URI for the remote Sentry this authentication policy will use.
Cache Timeout	How long Sentry will cache the results of the Sentry Authentication call to validate credentials. The cache value is in minutes, the default value is 30.
Cookie Name	The name of the cookie set by Sentry and returned to the client.
Host Header	The optional Host Header set by Sentry.
Propagate client Host header	This check box allows the Host header from the inbound request to be sent to the REST authentication remote server.

Sentry Authentication polices are similar to LDAP or SiteMinder Policies in that they will show up as Groups under the User ACLs. These policies can be used for either run time users or Design time users (WebAdmin access).

CUSTOM AUTHENTICATION

Custom Authentication policies allow Sentry authenticate the user in a custom manner using task list processing. Custom authentication could include such operations as making a database query with existing user attributes. Based on the success or failure of the request processing tasks, the authentication will succeed or fail. Under the scenario above, if the database query failed or some checks after it failed, the user would not authenticate. If identity attributes need to be added to the user's session, this should be done in the request processing tasks.



Custom Authentication Policy Configuration

When configuring your Custom Authentication policy, please consider the following:

FIELD NAME	DEFINITION
CUSTOM AUTH POLICY	
Policy name	The identifier for this Custom Authentication policy.
Enable privileged access	• With **Yes** selected, the user has access to the WebAdmin as a super user. • With **No** selected, the user still have access to the WebAdmin with all Domain privileges set for the Group which this user is a member of.
Restrict Menus	If using this policy for WebAdmin users, you can restrict menus for the users.
Role Policy	If using this policy for WebAdmin users, you can specify a defined Role for the users.
Request Processing	Sets up processing of the request so it can be structured properly for the Custom authentication server.
Response Processing	Sets up processing of the Custom Authentication server response to perform actions like converting entries in the response into user attributes.
Cache Timeout	How long Sentry will cache the results of the REST call to validate credentials. The cache value is in minutes, the default value is 30.
Cookie Name	The name of the cookie set by Sentry and returned to the client.
User Name Parameter	Identity attribute into which the username will be stored for addition to the user session. This is done via mapping tasks in the request processing.
Password Parameter	Identity attribute into which the user's password will be stored for addition to the user session. This is done via mapping tasks in the request processing.
Sha1 Parameter	Identity attribute into which the SHA1 hashed password will be stored for addition to the user session. This is done via mapping tasks in the request processing.
Cookie Name Parameter	Identity attribute into which the cookie name will be stored for addition to the user session. This is done via mapping tasks in the request processing.
Cookie Value Paramter	Identity attribute into which the cookie value will be stored for addition to the user session. This is done via mapping tasks in the request processing.
X.509 Certificate Parameter	Identity attribute into which the user's X.509 certificate will be stored for addition to the user session. This is done via mapping tasks in the request processing.
DN Parameter	Identity attribute into which the user's DN will be stored for addition to the user session. This is done via mapping tasks in the request processing.
Email Parameter	Identity attribute into which the user's email address will be stored for addition to the user session. This is done via mapping tasks in the request processing.

Custom Authentication polices are similar to LDAP or SiteMinder Policies in that they will show up as Groups under the User ACLs. These policies can be used for either run time users or Design time users (WebAdmin access).

USER POLICIES

Administrators may add, edit / view and delete users as well as limit the display of User policies. Users are immediately enabled upon creation. During the edit operation, assign users to groups or view user policy details. You may have an unlimited number of User policies on the product, limited only by disk space.

User Names

User names must be unique, are case sensitive, and may be from 1 to 80 alphanumeric characters. The '@' character, underscores, dashes and spaces are allowed; however no leading or trailing spaces are allowed.

User Passwords

A user password is a secret passphrase known only to the user for validating the user during login. Your Administrator initially enters a password for the user. User passwords must be unique, are case sensitive, may be from 6 to 255 alphanumeric characters, and may be any keyboard character.

Recoverable User Passwords and Password Hashes

User Passwords and their associated password hashes may be set for storage and retrieval under specific feature configurations. When the **Store recoverable passwords as well as password hashes** checkbox is checked, a copy of the cleartext password is stored on the system.

User Passwords for Users Assigned to the SNMPMonitor Group

User passwords for users who will be assigned to the SNMPMonitor group must be created in cleartext, are case sensitive, may be from 8 to 255 alphanumeric characters, and accepts the '@' character, underscores and dashes.

Additionally, the **Store recoverable passwords as well as password hashes** checkbox must be checked. When checked, a copy of the cleartext password is stored on the system.

Note: Before creating a user policy, consider which type of group the user will be assigned membership, and tailor the user password accordingly. For more information on SNMP Management, refer to the *Forum Systems Sentry™ Version 9.1 Monitoring and Reporting Guide*.

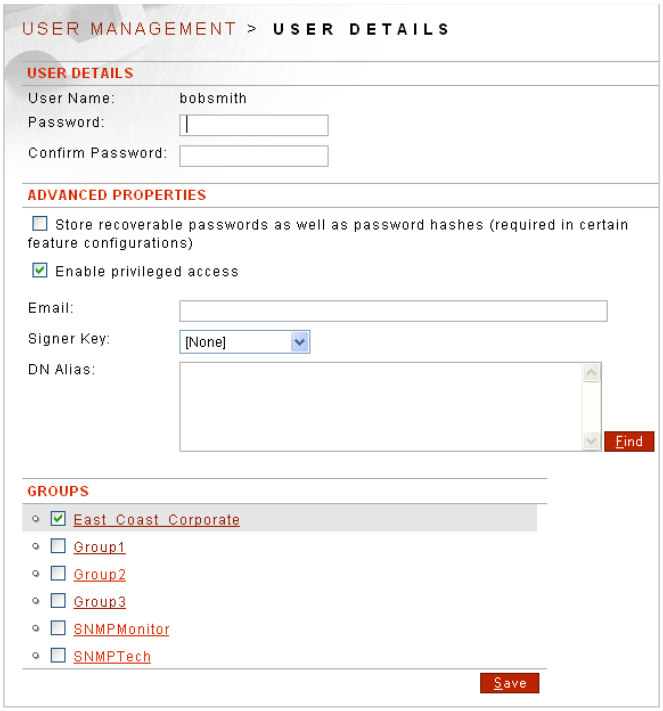
Users Management Screen Terms

When working in the USER MANAGEMENT screen, consider the following:

FIELD NAME	DEFINITION
User Name	A user name is the login name that uses of the system will use to gain access to the CLI, WebAdmin UI, the Policy Server machine, an Agent machine or to manage SNMP.
Password	Enter the user's secret passphrase in this field.
Confirm Password	Re-enter the password for the user in this field.
Store recoverable passwords as well as password hashes	When checked, passwords and their hashes are stored for retrieval and a copy of the cleartext password is stored on the system.
User Column	A list of existing users on the system.
Enabled Column	• **Green** status light indicates that this user is enabled. • **Red** status light indicates that this user is disabled. • **Yellow** status light indicates that a required functional element of this policy is disabled.
Type Column	There are two types of user policies: Forum System user and LDAP user. • The Forum System user is a user that is entered manually on the User screen or through the CLI. • The LDAP user is a user that is dynamically populated on the User screen as a result of fetching data from an LDAP server. For more information, refer to the LDAP screen.
DN Column	The Distinguished Name (DN) column displays a listing of DNs for LDAP users only.

Superusers

A privileged user (Superuser) is an Administrator user who has read and write access to all policies across all Domains. To set the privileged user attribute on a user, check the **Enable privileged access** checkbox in the User details screen of the product. For more information, refer to the Enable Privilege Access to a User Policy section.



Caveats with Superusers

When working as a Superuser in the system, consider:

- One Superuser cannot restrict another Superuser.
- Any Superuser user can restrict a non- Superuser.
- The system always protects against removing all superusers ensuring that there must remain at least 1 superuser at all times.

Assign Users to Groups

Users are assigned to groups via two methods:

- by editing a user from the Users screen (the USER DETAILS screen).
- by editing a group from the Groups screen (the GROUP DETAILS screen).

Forum System users may be assigned membership to:

- any user-defined group.
- the SNMPMonitor Group (system group)

Note: For information on the SNMPMonitor Group, refer to the *Forum Systems Sentry™ Version 9.1 Monitoring and Reporting Guide*.

Assign Group Membership to a User

The following graphic displays assigning a User membership to a Group from the USER DETAILS screen:

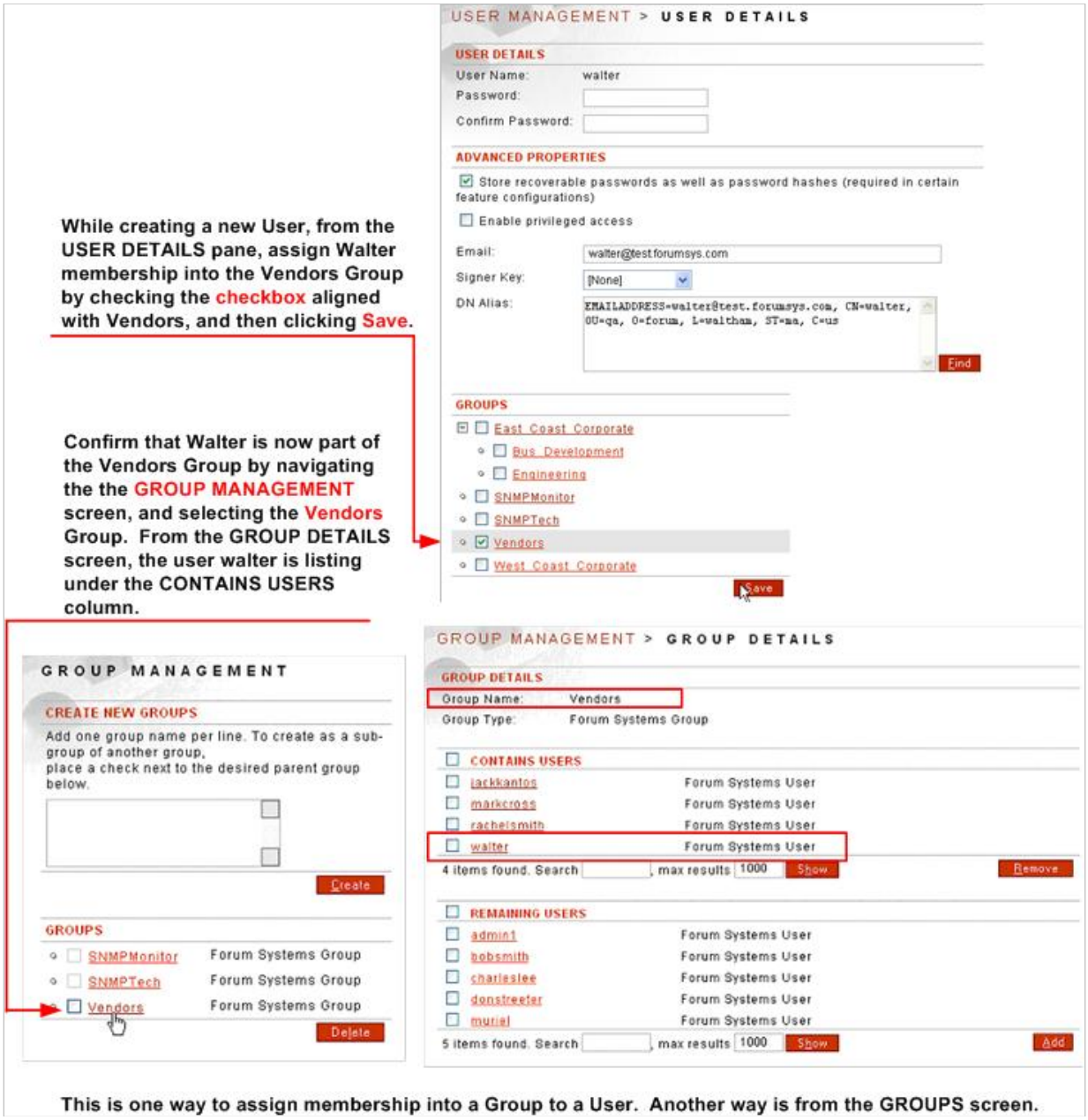


Figure 6: Assign Group Membership to a User.

Distinguished Name Attributes

The Distinguished Name (DN) attribute allows Administrators to identify a user or resource (computer name) and an associated value. DN aliases are made up of one or more DN attributes with the = character followed by a value.

Note: Forum Systems strongly recommends allowing the system to capture a User's DN alias by using the **Find** command from the USER DETAILS screen.

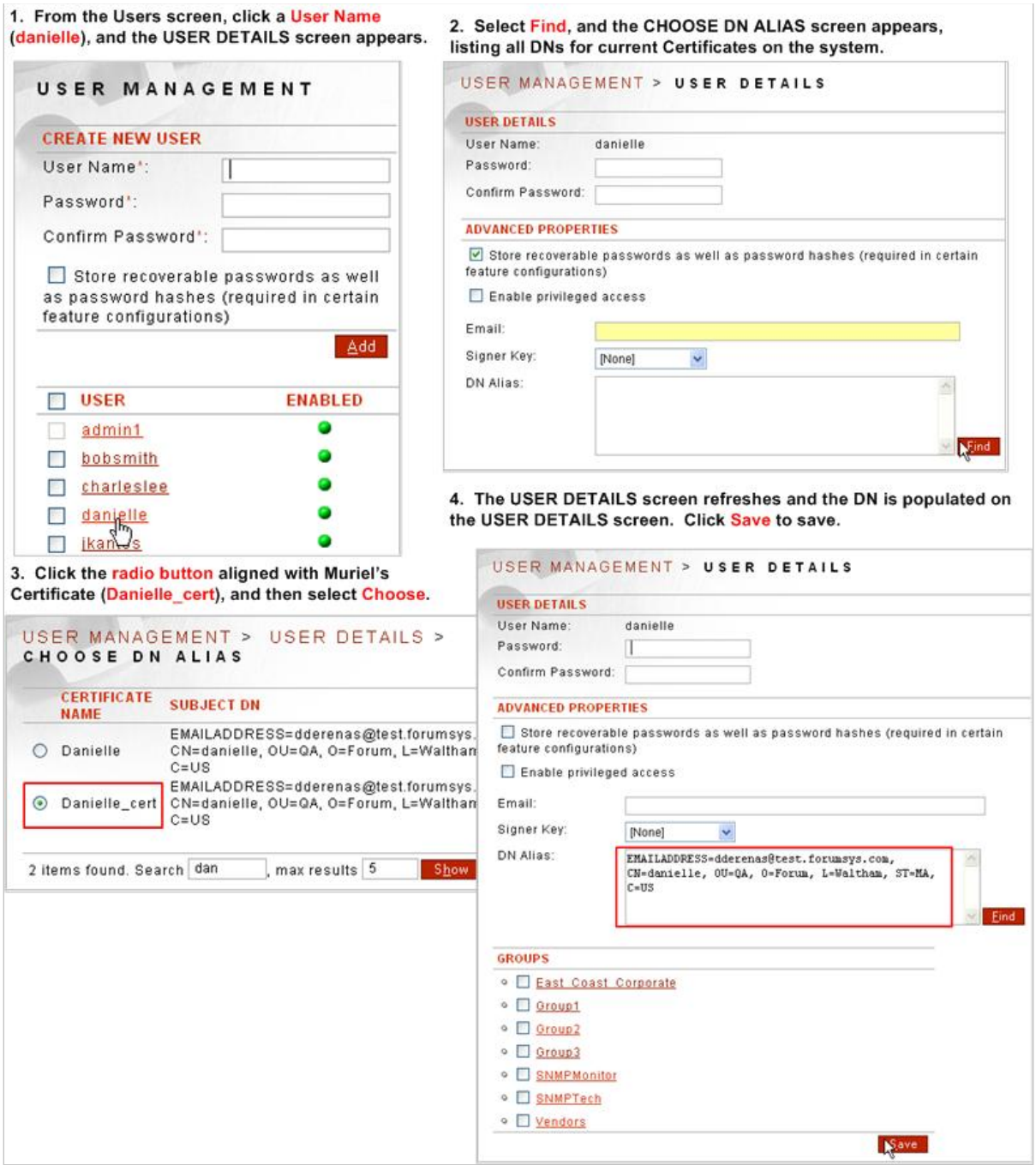
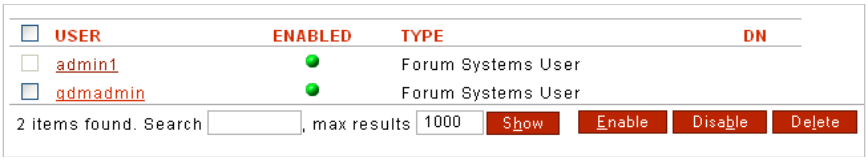


Figure 7: The System Retrieves DN Attributes for Users.

Restricted Self-Deletion

To prevent the current user from deleting his/her own user policy, the system automatically restricts self-deletion by displaying a grayed (disabled) checkbox aligned with the current user's name, admin1.



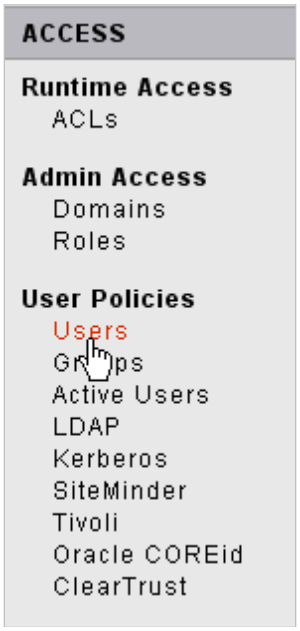
User Policies Examples

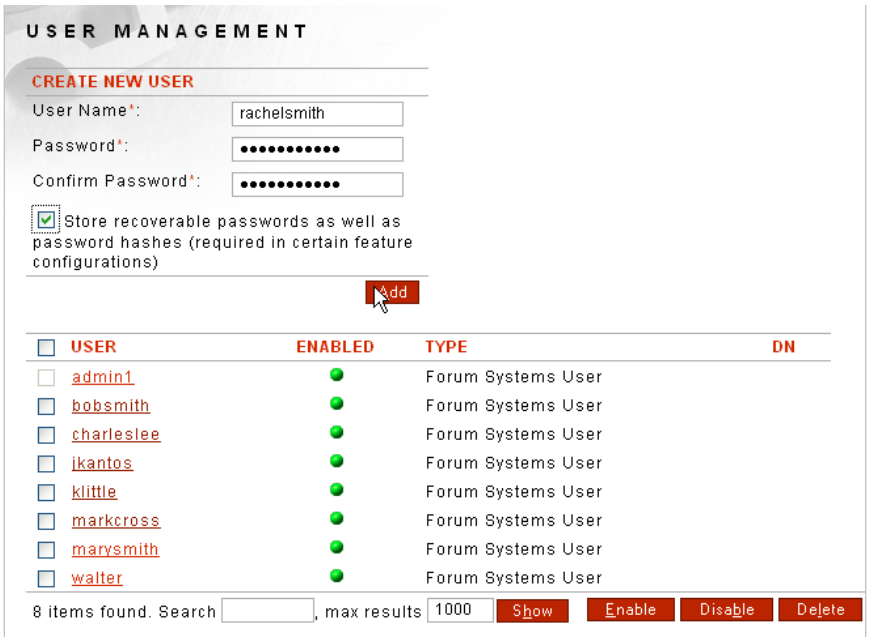
Examples for User policies include:

- Add User Policy.
- Assign Membership of a User to a Group.
- Enable Privileged Access to a User Policy.

Add User Policy

Follow these steps to add a User policy:

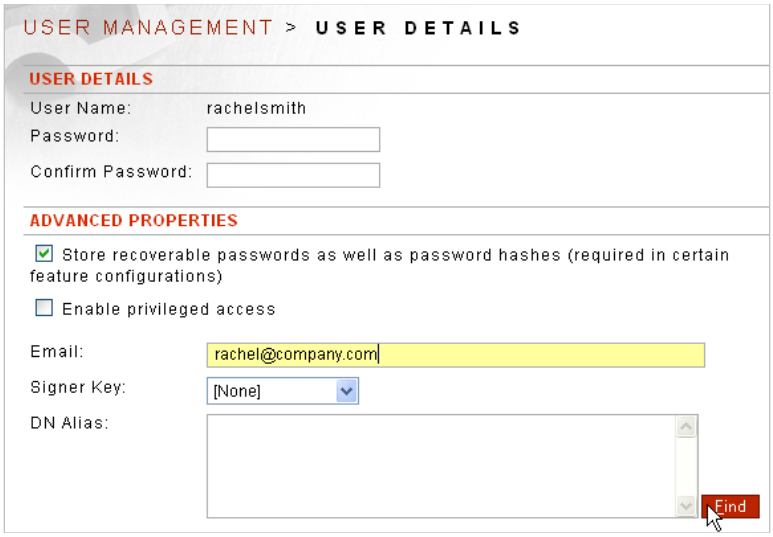


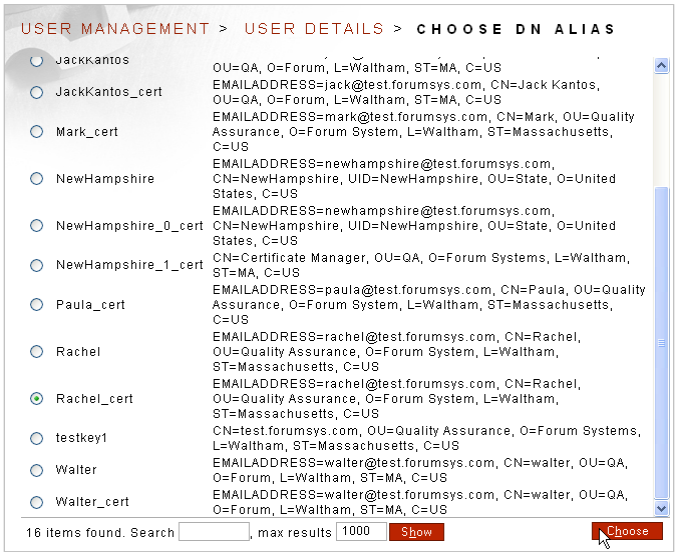


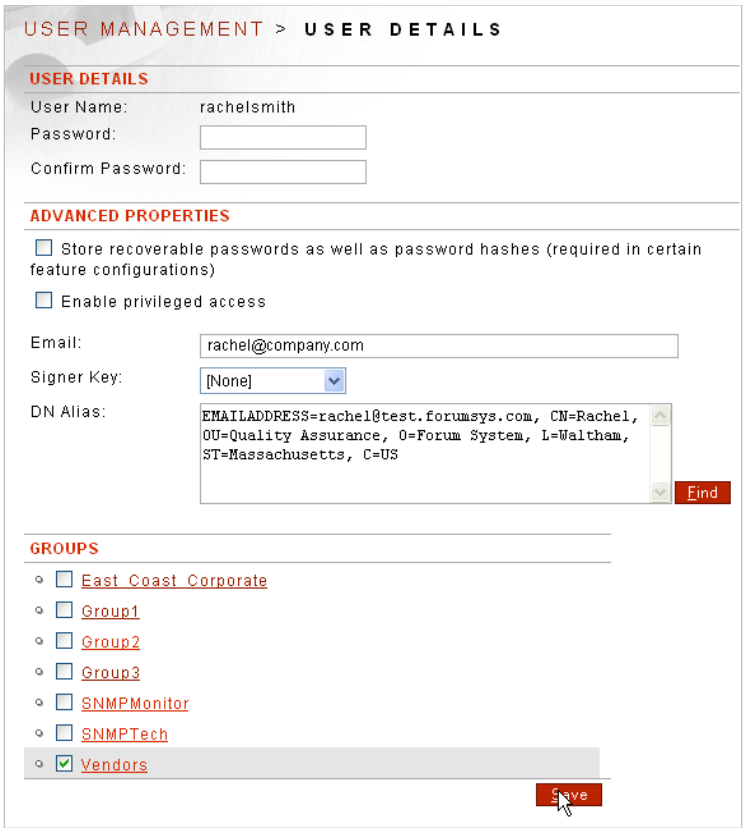
- Navigate to the **Users** screen.
- In the User Name field, enter a **User Name**.
- In the Password field, enter a **Password** for this user.
- In the Confirm Password field, re-enter the **Password** for this user.
- To store recoverable passwords, check the **Store recoverable passwords as well as password hashes** checkbox, and then click **Add**.
- The USER MANAGEMENT screen refreshes with new user name aligned with green status lights, indicating that the User Policy is enabled.

Adding Email and DN Alias for User

This instruction assumes that Rachel Smith's public certificate was previously added to the system.







- Click a **User Name**.
- Enter an **Email address** in the Email field, and then click **Find**.
- The CHOOSE DN ALIAS screen appears.
- Check the **radio button** aligned with the appropriate User Certificate, and then click **Choose**.

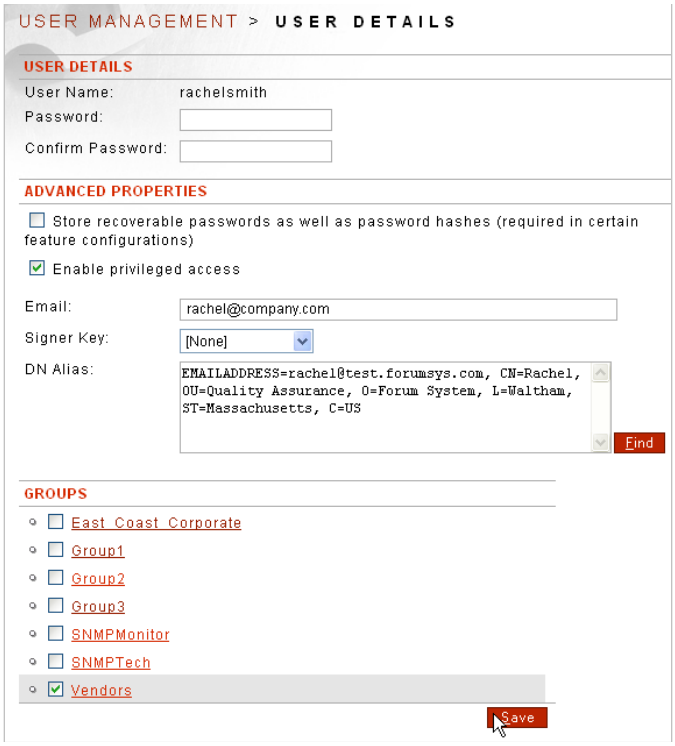
Continue with the next topic.

Assigning Membership of a User to a Group

- Check the **checkbox** aligned with the Group Vendors, and then click **Save**.

Continue with the next topic.

Enabling Privileged Access to a User Policy



- From the **Users** screen, select a **User Policy name**.

Note: When Rachel Smith was created earlier, the Store recoverable passwords as well as password hashes checkbox had been checked. This setting is visible on the USER DETAILS screen.

- Check the **Enable privileged access** checkbox, and then click **Save**.

ACTIVE USERS POLICIES

The Active Users screen provides a listing of all users currently on the product.

Active User Policies Examples

Examples for Active User policies include:

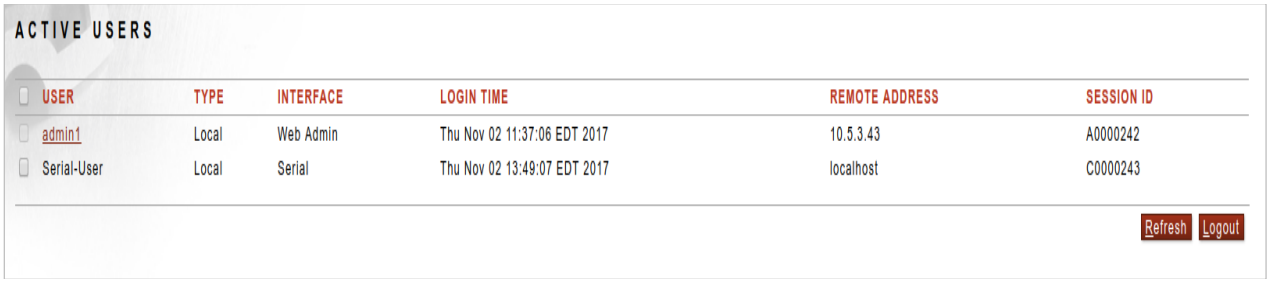
- Refresh Active Users.
- Logout Active User.

Refresh Active Users

To refresh the listing of Active Users, from the ACTIVE USERS screen, select **Refresh**. The ACTIVE USERS screen refreshes.

Logout Active Users

To logout an Active User, from the ACTIVE USERS screen, check the **checkbox** aligned with the user to logout, and then select **Logout**. The ACTIVE USERS screen refreshes.



Active Users Screen Terms

When working in the Active Users screen, consider the following:

FIELD NAME	DEFINITION
User	The name of the user who is currently active on the system.
Type	Local active users who are users created on the system or users who are authenticated via LDAP, Tivoli, SiteMinder, etc.
Interface	• WebAdmin means that the active user is currently on the WebAdmin interface. • Serial means that the active user is currently connected to the CLI via serial interface. • SSH means that the active user is currently connected to the CLI via SSH.
Login Time	Timestamp when this active user logged in to the system.
Remote Address	The IP address of the computer used to login to the system.
Session ID	Session ID which will be displayed next to this user's activities in the Audit log. JFW

GROUP POLICIES

Administrators may add, edit and delete groups, view group details and limit the display of members of a Group policy. Groups are immediately enabled upon creation. During the edit operation, assign users to groups or view group policy details. Groups are collections of users on the system. You may have an unlimited number of Group policies on the product, limited only by disk space.

There are three types of groups:

- User-defined groups
- System groups
- Identity Policy groups (LDAP, Active Directory, Siteminder, REST, etc)

When working with groups, consider that:

- A group represents zero to many users, security principals or resources in the system.
- User-defined groups and Identity Policy groups are ultimately assigned membership to Access Control Lists (ACLs) to grant or restrict access to resources
- The system group SNMPMonitor cannot be deleted.
- Group policy names must be uniquely named at each group level.
- Enter only one group name per line.
- If the creation of a group name fails, the Administrator will have to re-enter the data.

User-defined Groups

Administrators may:

- create one or more User-defined groups on the Groups screen.
- edit a group and subsequently add existing users to the group.
- delete a User-defined group at any time.

Note: Deleting a User-defined group does not delete the User policies of the users that were previously assigned membership to that group.

SNMPMonitor System Group

The SNMPMonitor group is a system group that cannot be deleted. This system group is prefaced by a gray selection box, preventing deletion. This group is used for users who will be associated to SNMP monitoring policies on the device.

Previous releases of Sentry also included the SNMPTech system group. This was an unnecessary duplicate of the SNMPMonitor group and has been removed. Any users that were previously in this group should be added to the SNMPMonitor group.

Note: For more information on SNMP management, refer to the *Forum Systems Sentry™ Version 9.1 Monitoring and Reporting Guide*.

Identity Policy Groups

Identity Policy groups represent the membership of users who are able to dynamically authenticate to the policy at runtime (transactions) or at log-in time (administrators).

Assign Users to Groups

Users are assigned to groups via two methods:

- by editing a user from the Users screen
- by editing a group from the Groups screen

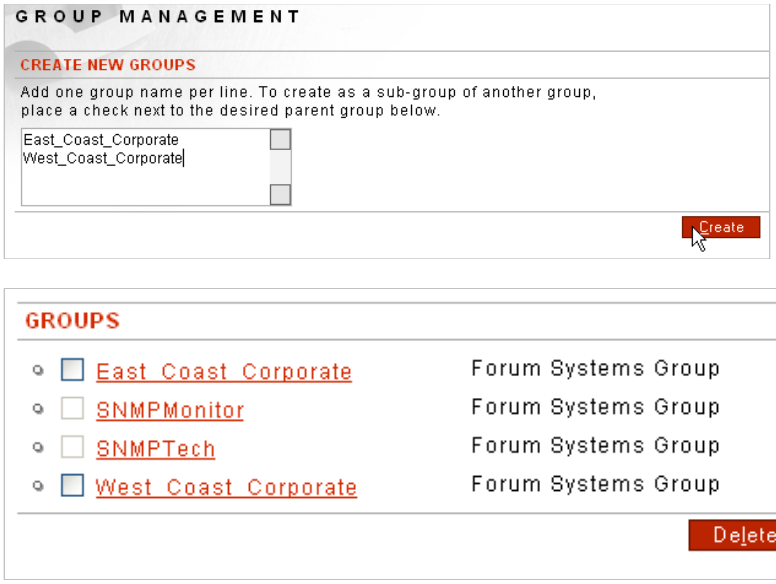
Group Policies Examples

Examples for Group policies include:

- Add Group Policy.
- Assign Users Membership to a Group.

Add Group Policy

Administrators may add new groups individually or in bulk. Adding groups in bulk is accomplished by entering group names, one per line, in a text box. Administrators may add multiple groups, assign users, and finally edit each group as users are assigned to them. This example displays adding multiple groups at one time.

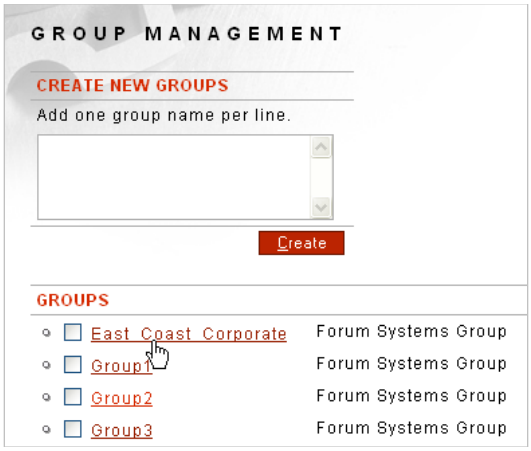


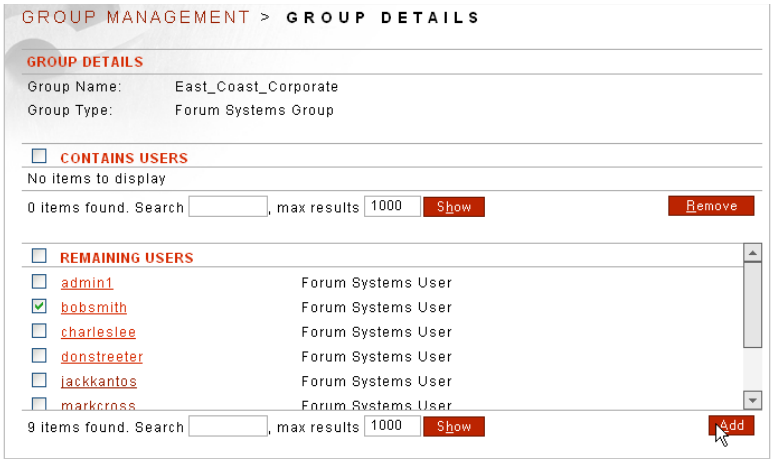
- Navigate to the **Groups** screen.
- In the text box, enter one **Group Name** on each line of the text box.
- Click **Create**.

Note: After creating a Group, click off the Groups screen, then return to view a refreshed and re-alphabetized listing of group policies.

Assign Users Membership to a Group

Follow these steps to assign one or more Users' membership to a Group:





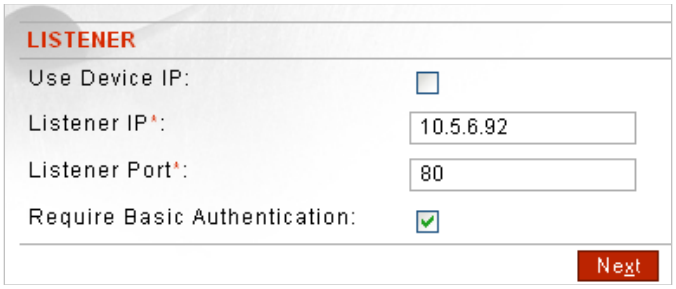
- Navigate to the **Groups** screen.
- Click a **Group Name** and the GROUP DETAILS screen appears.
- Check the **checkbox** aligned with one or more Users, and then click **Add**.

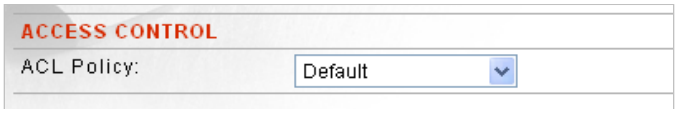
ACCESS CONTROL LISTS AND SECURITY

Access Control is managed by User Access Control Lists (ACLs) on the system. When a document comes into the system, the system checks for access control at three separate points. At each point, access control is triggered by a User ACL. Access Control may be set for a user at any one, two or three points. When a message comes into the system:

Network Listener Level

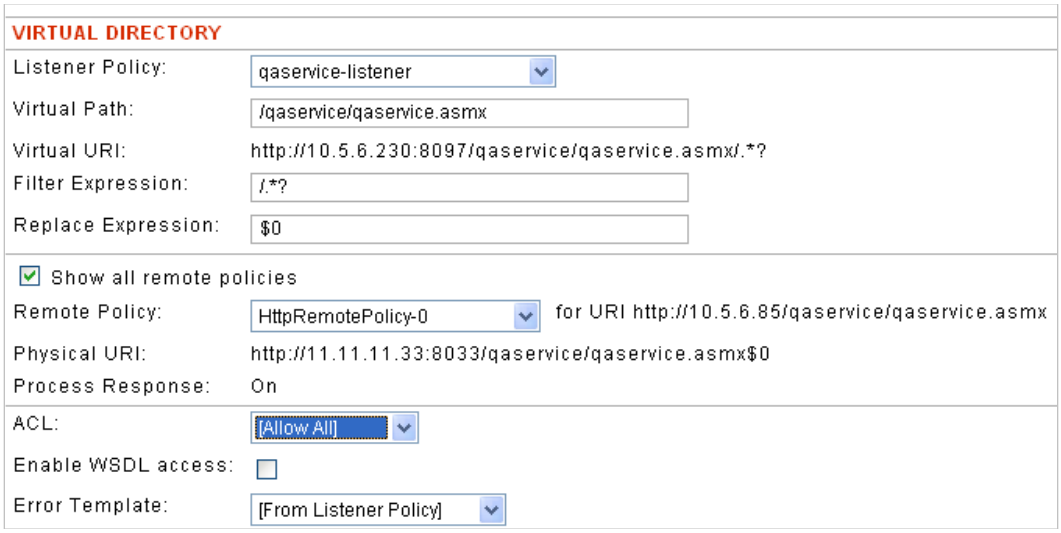
The first security check occurs at an HTTP Listener Network policy, which is associated with a WSDL policy, where access control is managed by both the check in the **Require Basic Authentication** checkbox and a selected **ACL** under Access Control. Only users whose groups are associated with an ACL are granted access to this listener.





Port Level

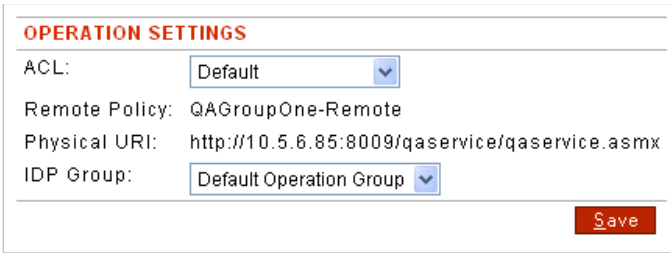
The second security check occurs at the Service port on a Virtual Directory of a WSDL policy, where access control is managed by a selected **ACL**.



Selecting Allow All as the ACL policy means that the user is identified from the protocol or the document and is matched to a known user in Forum or in a third party user store. The user is not restricted by any Forum ACL.

Operation Level

A third security check can be configured to occur at the Operation level of a WSDL policy, where access control is managed by a selected ACL. An ACL may be associated from one to all the operations of a WSDL policy.



The only user who has access all the way through a WSDL policy at the Service and Operation level is a user whose group is associated with the ACL selected on the WSDL policy.

Each user may be a member of one or more groups. Each group may be associated with one or more ACLs. LDAP users are defined by the criteria specified on the dynamic LDAP policies. The LDAP Policy itself is treated as a Group and may have access to ACL attribute privileges.

In the following graphic, ACL1 includes Group1, Group2 and Group3.

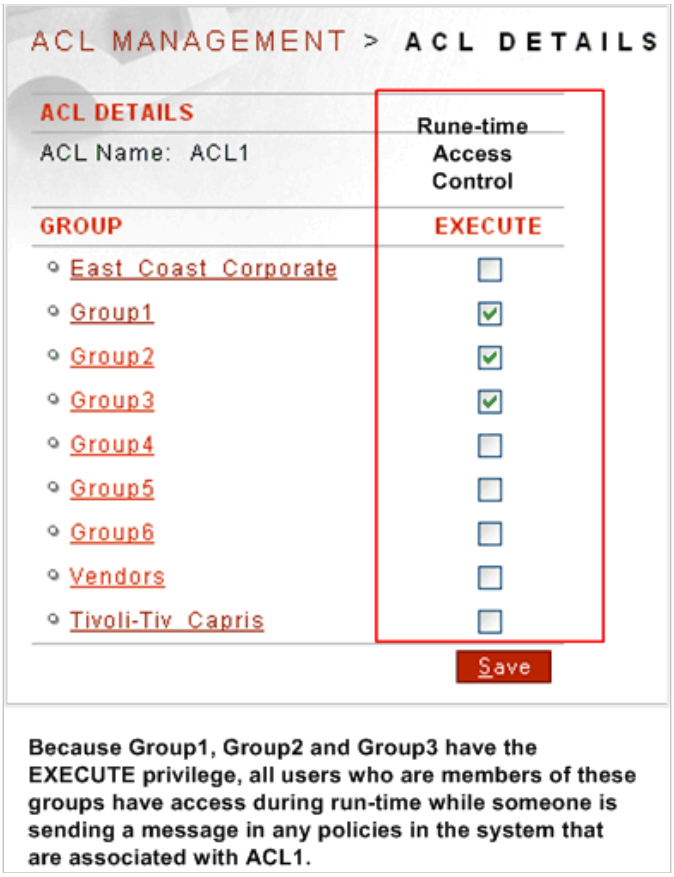


Figure 8: Run-time Access Control.

IP ACL POLICIES

IP Access Control List policies provide a method of defining a global set of IP ranges that can be applied to HTTP/S and SMTP listener policies. Users may define one or more IP addresses that are either allowed or denied on the listener.

About the Unrestricted IP ACL

The factory system default IP ACL policy, **Unrestricted**, cannot be edited and cannot be deleted. This IP ACL policy allows all IP ranges.

IP ACL Screen Terms

When working in the IP ACL Policies screen, consider the following:

FIELD NAME	DEFINITION
Policy Name	The identifier for the IP ACL policy.
IP Ranges	A listing of IP addresses to be allowed or denied to connect to the listener, i. e.: 172.16.1-172.16.1.33 192.168.4.33 Leaving this field blank allows all client IPs to connect to the listener.
Allow or Deny	• With **Allow** selected, the defined set of IP address ranges are allowed on this policy. • With **Deny** selected, the defined set of IP address ranges are denied on this policy.

Note: For users who might accidentally create an IP ACL policy that would deny them access to their network IP ranges, a system message appears stating that it cannot be done.

IP ACL Policies Examples

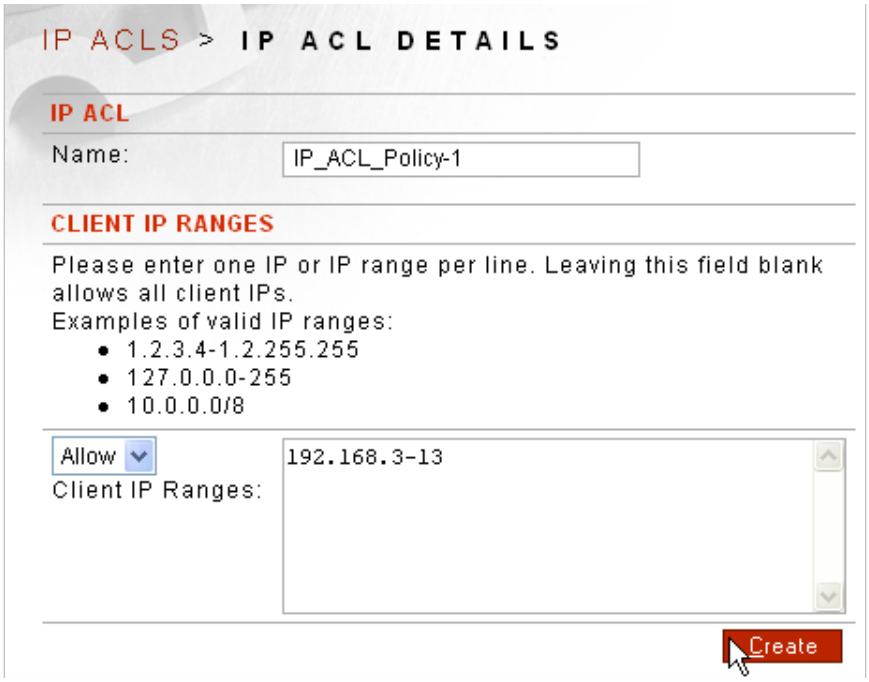
The example for IP ACL policies is:

- Add an IP ACL.

Note: For information on editing / viewing, or deleting an IP ACL policy, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9.1 Web-based Administration Guide*.

Add an IP ACL Policy

Follow these steps to add IP ACL policy.



- Navigate to the **IP ACLs** screen.
- On the IP ACL DETAILS screen, enter an **IP ACL Policy name** in the text field or accept the default name.
- Decide to retain the **Allow** or **Deny** option in the Client IP Ranges drop down list.
- Enter one or more **IP address(es)** or **IP range(s)** for the HTTP/S or SMTP client(s) that will use this policy.

- Click **Create**.

USER ACL POLICIES

User Access Control Lists (ACLs) are used to specify run-time access privileges that members of each Group has to XML policies, WSDL policies and Network policies on the WebAdmin. The system supports an unlimited number of User ACL policies, limited only by disk space. User ACLs apply only to run-time authentication through the Execute privilege.

User-defined groups and System groups may be assigned membership to User ACLs. Administrators may create, edit and delete User ACLs as well as view User ACL details. When working with User ACLs, consider that:

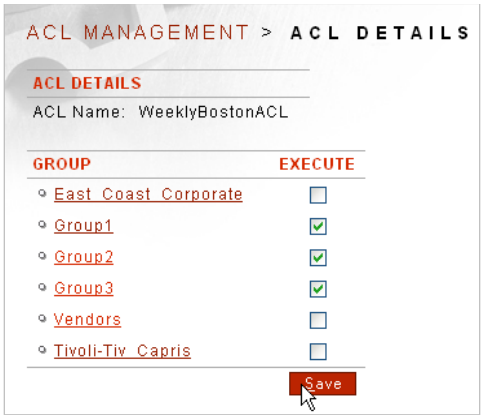
- A User ACL uses zero or more groups on the system to control user access.
- Each User ACL name must be unique.
- Enter only one User ACL name per line.
- Once created, User ACLs are listed alphabetically by name.
- The default User ACL, Default, is protected from deletion.

Note: Although User ACLs are created in the User ACLs screen, they may be applied in different ways. For more information, refer to the *Forum Systems Sentry™ Version 9.1 WSDL Policies Guide* or the *Forum Systems Sentry™ Version 9.1 XML Policies Guide*.

For information on editing / viewing, or deleting a User ACL policy, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9.1 Web-based Administration Guide*.

Restrict Run-time Access with User ACLs

User Access Control Lists are created in the User ACLs screen, and may be applied or associated with XML policies, WSDL policies and Network policies. Run-time access may be granted or restricted though User ACL Policies



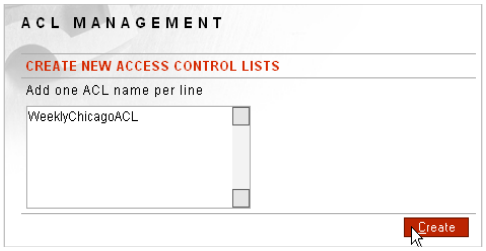
User ACL Policies Examples

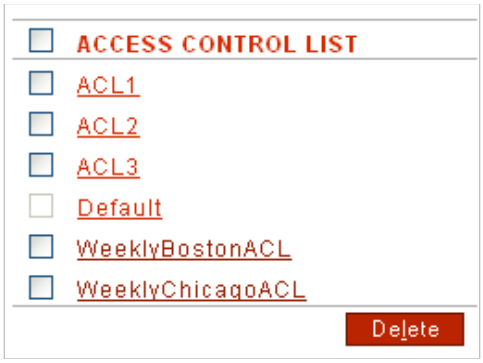
The example for User ACL policies is Add a User ACL and Associate Groups to a User ACL.

Add an ACL Policy

Administrators may add Access Control Lists individually or in bulk.

Adding the ACL



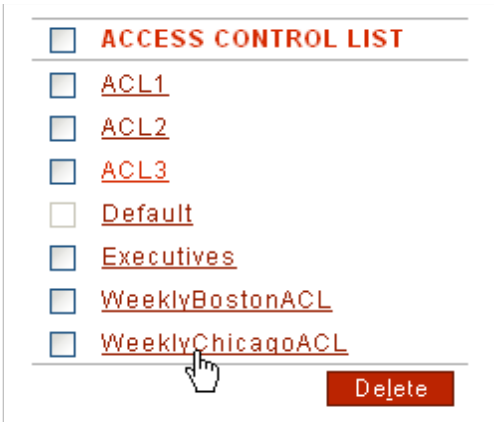


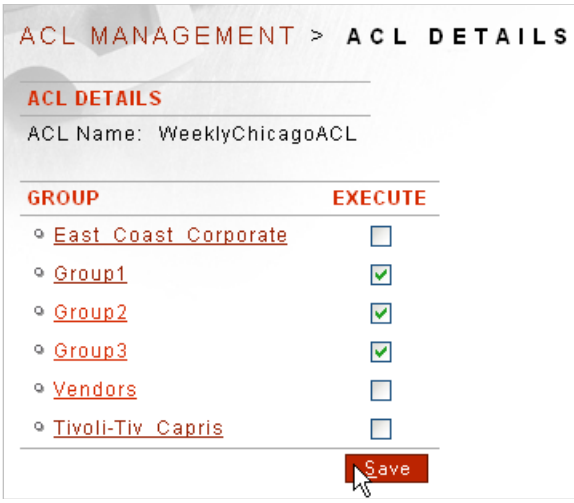
- Navigate to the **ACLs** screen.
- From the CREATE NEW ACCESS CONTROL LISTS section of the screen, in the text field, enter **one ACL name** on each line.
- Click **Create**.

You have created two ACLs and each contains all the existing groups. To assign groups to an ACL, you must edit the ACL Policy.

Continue with the next topic.

Assigning Groups to an ACL





- From the **ACLs** screen, click an **ACL Name**.
- Check the **EXECUTE** checkbox for each group whom you wish to grant this privilege to.
- Click **Save**.

XACML POLICIES

eXtensible Access Control Markup Language (XACML) policies define a declarative access control policy language implemented in XML and a processing model describing how to evaluate access requests according to the rules defined in the access policies. XACML is meant to promote common terminology and interoperability between access control implementations by multiple vendors. XACML is an Attribute Based Access Control (ABAC) system where attributes associated with a user are inputs into the decision of whether a given user may access a given resource or service in a particular way.

Forum Sentry enables the ability to generate XACML client requests to a target XACML enabled identity provider in order to provide access control decisions based on the target credential being evaluated.

XACML adoption is sparse in the industry and is not commonly seen in the modern approaches to identity management for API or SOA initiatives as it has been mostly superseded by the preferred standards SAML and OAuth. However, there remain some XACML providers in the industry. Forum Sentry supports the a common open source implementation of XACML via the OpenSSO option on the XACML policy which provides native OpenSSO API support which performs similar to XACML with some proprietary aspects of the OpenSSO API. Alternatively, for standards-based XACML 2.0 interfaces, the XACML 2.0 option is provided.

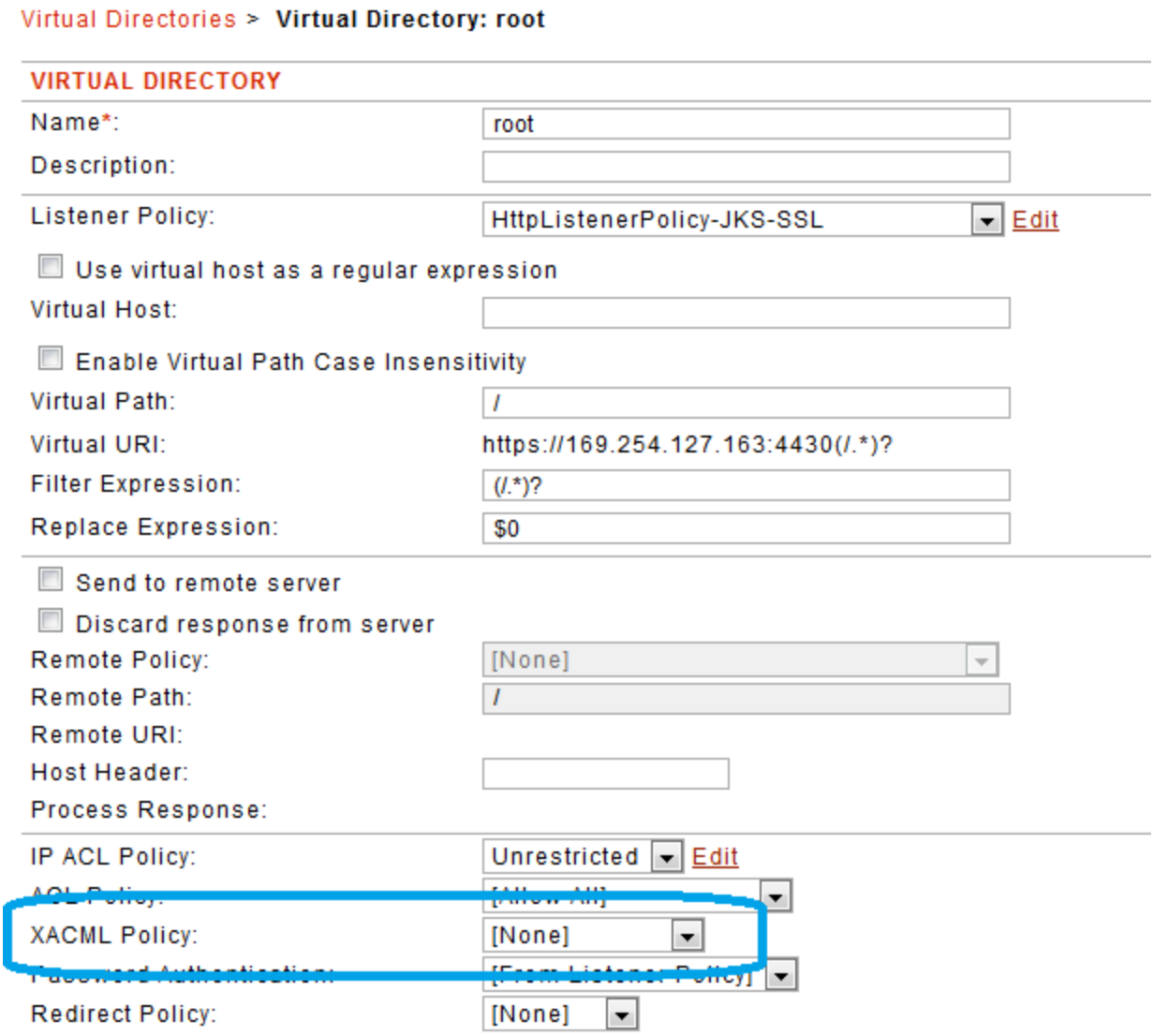
XACML Policy Screen Terms

When working in the XACML screen, the following are the relevant field definitions

FIELD NAME	DEFINITION
Policy Name	The name of the XACML Policy. This is used to reference the policy from the policies that consume XACML policies
Compatibility	Choose XACML 2.0 for standards-based XACML, or OpenSSO for proprietary native integration to the Oracle OpenSSO XACML implementation
Remote Policy	The Sentry Remote Policy that defines the protocol settings to use to communicate with the target XACML Identity Provider
Remote Path	The remote path to use when making XACML requests over HTTP(S)
Cache Timeout (mins)	Setting used to cache XACML response to reduce the network I/O latency and redundant calls to the XACML identity provider. Set this value to 0 to disable the cache
SAML Issuer	The URI that is used to designate the SAML issuer attribute of the XACML client request.

Applying XACML Policy

XACML policies are associated in the same locations that ACL policies can be associated, on the Content Policy. An example of the location is shown in the figure below.


Virtual Directories > Virtual Directory: root
VIRTUAL DIRECTORY
Name*:
root
Description:
Listener Policy:
HttpListenerPolicy-JKS-SSL
Edit
Use virtual host as a regular expression
Virtual Host:
Enable Virtual Path Case Insensitivity
Virtual Path:
/
Virtual URI:
https://169.254.127.163:4430(/.*)?
Filter Expression:
(/.*)?
Replace Expression:
$0
Send to remote server
Discard response from server
Remote Policy:
[None]
Remote Path:
/
Remote URI:
Host Header:
Process Response:
IP ACL Policy:
Unrestricted
Edit
XACML Policy:
[None]
Redirect Policy:
[None]

OVERVIEW OF MULTI-DOMAIN ADMINISTRATION

Multi-domain administration is a feature provided by the product that gives the ability to have one or more sets of administrators on the system. The administrator sets can be configured to have separate and distinct views of policies and configuration settings. This allows multiple sets of Administrators to share resources of the system and only see polices applicable to their administrator set.

The multi-domain administration features are managed through Domain policies on the system. Domains provide settings for read and write attributes. These attributes are applied to the groups defined on the system. An Administrator would obtain the corresponding attributes of read and write based on their membership to group(s). Each Domain policy represents a separate administrator set corresponding to a distinct and separate view of policies and settings.

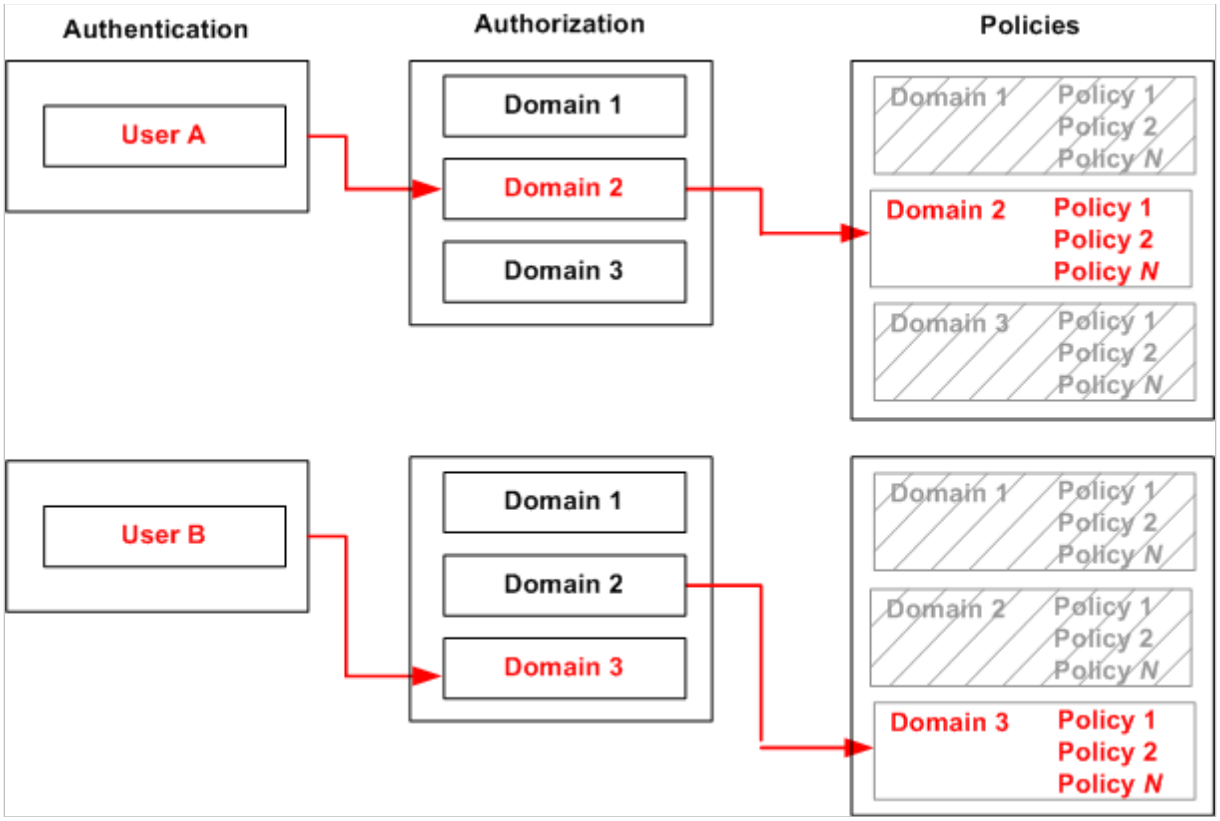


Figure 9: Domains Manage Multi-domain Administration.

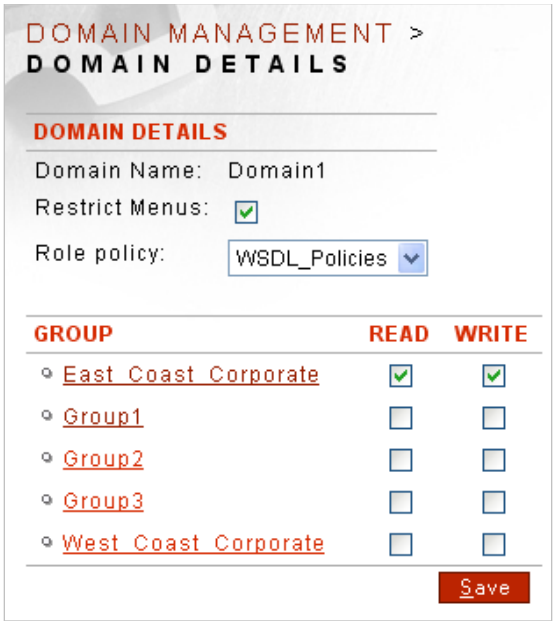
Each of the Domains shown in the above graphic represents three distinct sets of Administrators. When an Administrator performs a login, the user is checked against its group memberships, and subsequently the group memberships are checked against the Domain settings. Upon access to the system, the Administrator will be able to view and create policies and settings specific to the groups who have membership in each distinct Domain.

If the Administrator is a member of more than one Domain, s/he will view the policies created in all the Domains that s/he has membership to.

When new policies are created on the WebAdmin, they are always associated with the Active Domain.

Administrator privileges are set on the DOMAIN DETAILS screen for a Domain policy. All groups defined on the system are shown for selection of Read and/or Write privileges. The Read and Write settings provide the groups with privileges to view, create, modify, and delete policies.

Note: For more information on the Read and Write privileges and hierarchical administration, refer to the *Forum Systems Sentry™ Version 9.1 Access Control Guide*.



The graphic above displays a Domain with group administration settings. Users who are members of the East_Coast_Corporate Group (with Read and Write privileges) may logon to the system and:

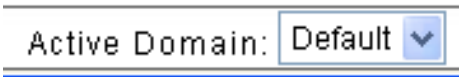
- Create
- View / Edit
- Rename
- Enable / Disable
- Delete any policies created by users in Domain 1.

DOMAINS

The Domains screen manages design-time access control through Read and Write privileges set on various Groups via a Domain Policy. Domain policies also provide a method of associating a Role Policy to a Domain, and therefore, to members of any Groups associated with a given Domain. Domains provide a method of separating business units so that each business unit has its own policies and privileges. Consider a domain as a sand-box where all policies defined are only allowed to be viewed and edited by administrators who are members of the domain. Other administrators in other domains do not see these policies when accessing the system.

Administration Domain

The Active Domain is displayed at the bottom of the WebAdmin UI. After installation of the system, the default Domain, labeled Default, is visible.



Domain Policies Examples

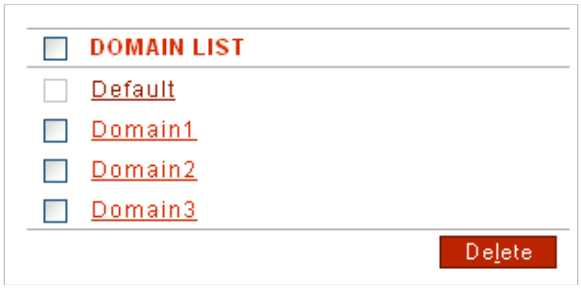
Examples for Domain Policies are:

- Add a Domain Policy.
- Assign Groups to a Domain.
- Associate a Role to a Domain.

Add a Domain Policy

Only superusers or WebAdmin users associated with a Role that grants access to domain Policies may add Domain Policies individually or in bulk.

Adding the Domain Policy

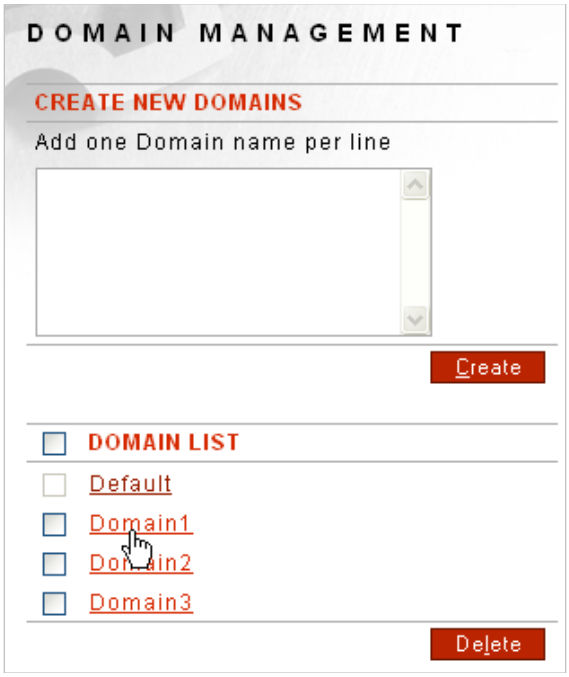


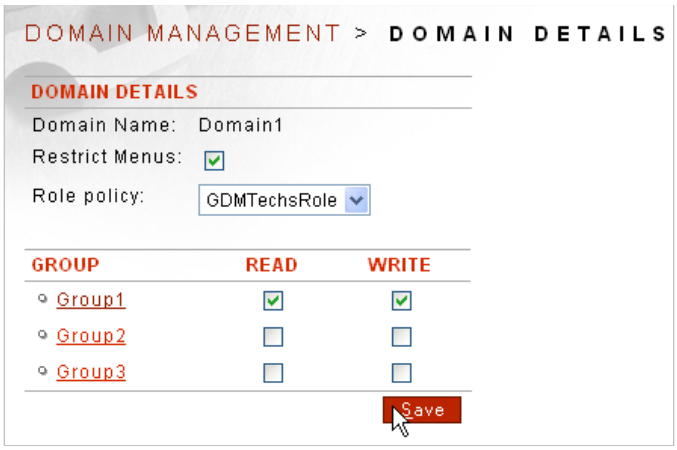
- Navigate to the **Domains** screen.
- Under CREATE NEW DOMAINS, in the text field, enter one **Domain Policy name** on a line.
- Click **Create**.

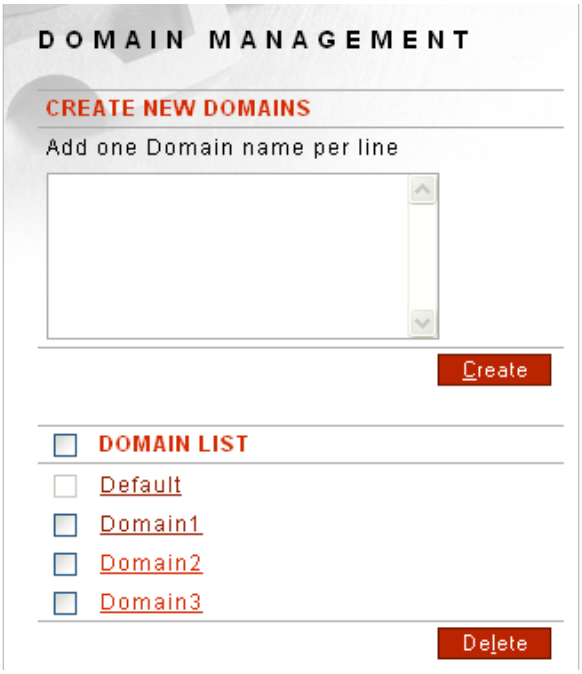
To assign a Group to a Domain or to enable a Role (Role List) on this Domain, you must edit this Domain Policy.

Continue with the next topic.

Assigning a Group to a Domain







- From the **Domains** screen, click a **Domain name** link.
- Check the **READ** and **WRITE** checkboxes aligned with a chosen Group name(s) for each privilege you wish to assign to this in this Domain Policy.

Associating a Role to a Domain

This instruction assumes you have already created a Role policy and populated it with, at minimum, one Menu.

- Under DOMAIN DETAILS, check the **Restrict Menus** checkbox to assign a Role Policy to this Domain Policy.
- From the Role Policy drop down list, select the **name** of a Role Policy to associate with this Domain Policy.
- Click **Save**.

ROLE POLICIES

The Roles screen allows the definitions of specific administrative roles to be defined. This is accomplished by designating the screen policies that can be viewed and edited when logging in. Designations of roles can hide certain menu items, and restrict menus to only the sub-set allowed based on the Administrative credentials

Role based policies can be defined within local groups, or based on Identity Policies where in each Identity Policy configuration screen there is an optional Role Policy configuration. If the Role policy configuration is enabled, the Identity Policy changes from a runtime policy to an administrator policy and enforces the Role of the administrator based on the credentials provided at login.

By creating a Role policy that contains specific Menu options, a superuser may restrict access of a user working in his/her Domain on the system. This feature allows for menu-level security throughout the system.

Each Role Policy may have from none to all Menu options assigned to it. The system supports an unlimited number of Role Policies.

The list of Allowed Menus for assignment to a Role Policy includes:

- ACLs
- Active Users
- Agent Groups
- Agents
- AmberPoint
- Archiving
- CA Antivirus
- Check Point
- ClearTrust
- Control
- CRLs
- Default AV
- Decryption
- Documents
- Encryption
- General Info
- Getting Started
- Groups
- Help
- IDP Actions
- IDP Blocking
- IDP Groups
- IDP Rules
- IDP Schedules
- Import / Export
- Internal Logs
- JMX Remote
- Keys
- LDAP
- DOMAINS
- Network
- Network Policies
- OpenPGP
- Oracle COREid
- Oracle WSM
- Pattern Match
- Preferences
- Remote Syslogs
- Roles
- Settings
- Signature
- Signer Groups
- SiteMinder
- SNMP
- SSL
- Statistics
- System
- Task Lists
- Templates
- Tivoli
- Unicenter WSDM
- Upgrade
- Users
- WS Monitoring
- WS Reports
- WSDL API
- WSDL Libraries
- WSDL Policies
- Verification
- XML Policies

By limiting the selection of Menu options allowed on a Role Policy, superuser may create re-usable lists for granting access to users and screens.

Earlier, when defining a Domain in the DOMAIN DETAILS screen, a superuser assigned one or more Groups of users to that Domain and which Role Policy that Group of users could access.

Role Policies Examples

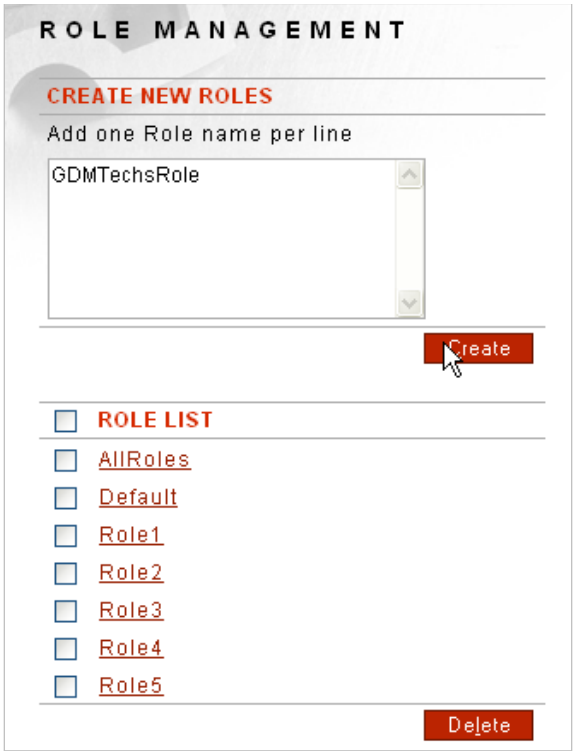
The examples for Role Policies are:

- Add a Role and Assign Allowed Menu Options.
- Delete a Role Currently in Use by the System

Add a Role Policy

Administrators may add Role Policies individually or in bulk.

Adding the Role Policy

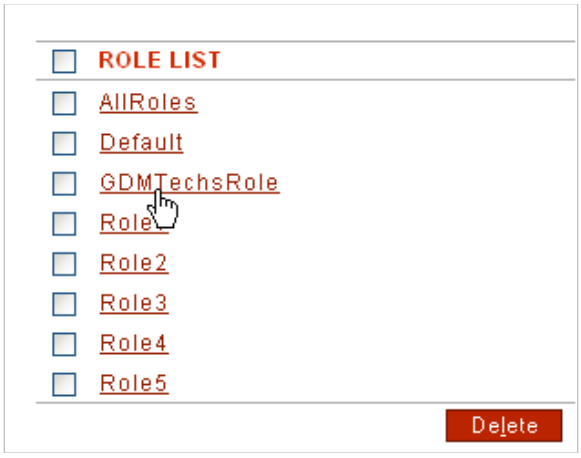


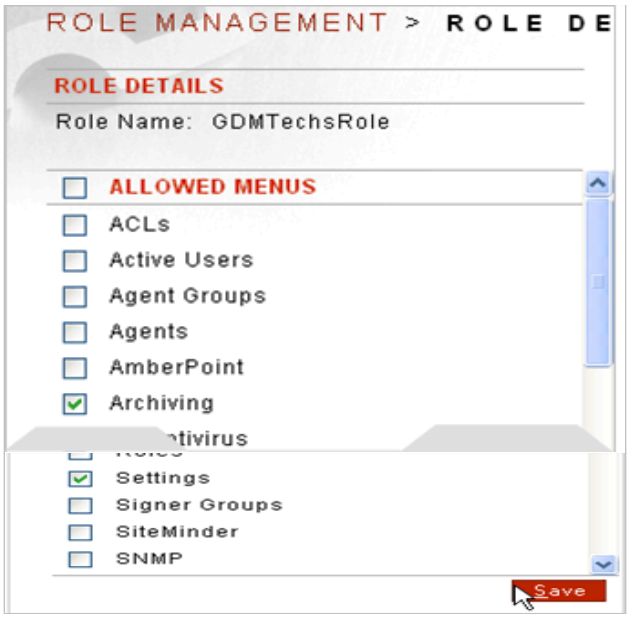
- Navigate to the **Roles** screen.
- Under CREATE NEW ROLES, in the text field, enter one **Role Policy** name on each line.
- Click **Create**.

To assign Menu Options to a Role Policy, you must edit the Role Policy.

Continue with the next topic.

Assigning Allowed Menu Options to a Role Policy



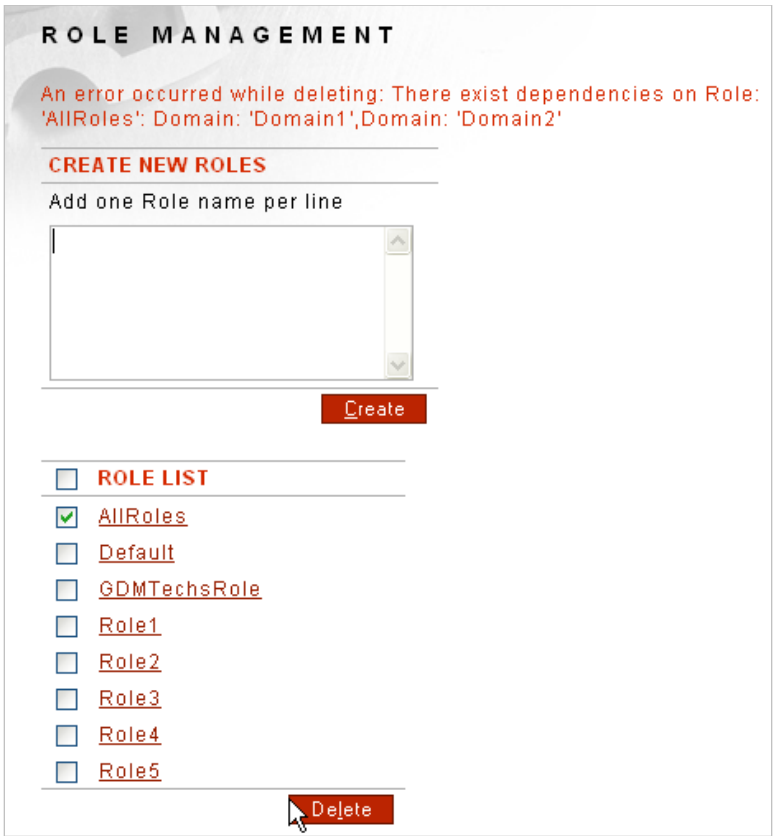


- From the ROLE MANAGEMENT screen, click a **Role name**.
- Check the **checkboxes** prefacing all Menu options that are allowed on this Role List.

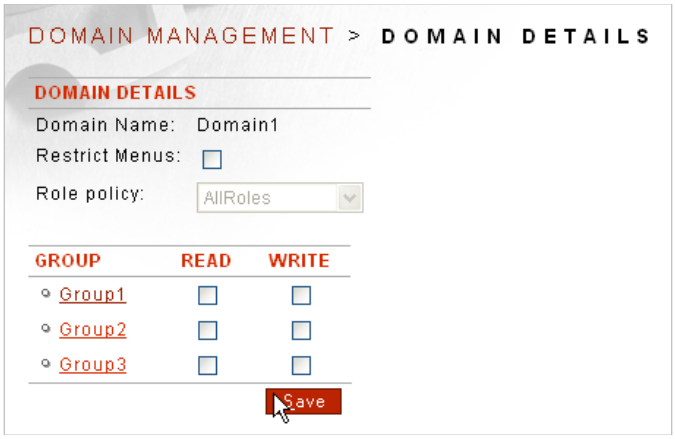
- Click **Save**.

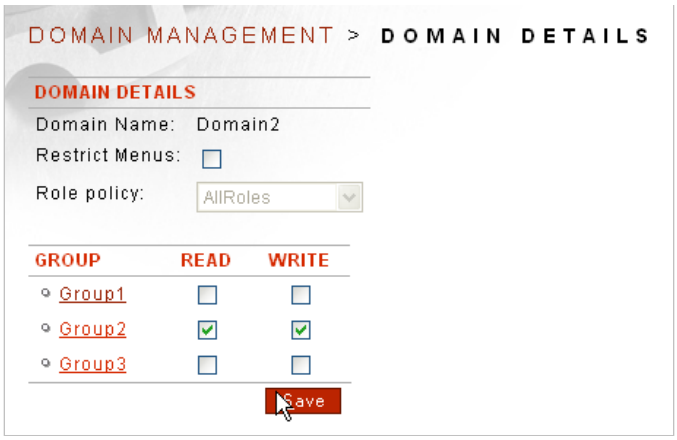
Delete a Role Policy Currently In Use by the System

If trying to delete a Role Policy that is currently in use by the system, the following message appears at the top of the Role Policy screen:

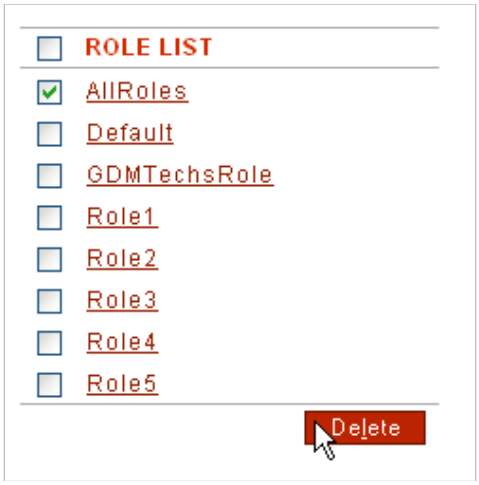


- Edit the referenced **Domain Policies,** uncheck the **Restrict Menus** checkbox, and select **Save**.





- Return to the **Roles** screen. Select the **checkbox** prefacing the Role Policy, and select **Delete**.



APPENDIX

Appendix A - Constraints in Access Control Guide

ELEMENT	CONSTRAINTS	CHAR COUNT
Group Policy Name	Unique per level and case sensitive. The “@”character, underscores, dashes and spaces are allowed; however, no leading and trailing spaces are allowed.	1-79
User ACL Policy Name	Unique and case sensitive. The “@”character, underscores and dashes are allowed.	1-79
LDAP Host name	Unique and case sensitive. May be from 1 to 32 alphanumeric characters, may include underscores, dashes and at least one period. However, no training or leading periods are allowed.	1 to 32
LDAP port	Integer between 1 and 65535.	1 to 5
Root DN	Root DN may be from 0 to 255 alpha characters, may include spaces, period and at least one equal sign. Multiple groupings must be separated by commas. Example: Root DN = ou=People, o=qa-lab.forumsys.com.	0 to 255
Username attribute on LDAP Policies screen	May be from 1 to 255 alphanumeric characters, semicolons and dashes are allowed. Spaces are not allowed.	1 to 255
Active Username attribute on LDAP Policies screen	May be from 1 to 255 alphanumeric characters, semicolons and dashes are allowed. Spaces are not allowed.	1 to 255
Group Attribute on LDAP Policies screen	May be from 1 to 255 alphanumeric characters, semicolons and dashes are allowed. Spaces are not allowed.	1 to 255
Password Attribute on LDAP Policies screen	May be from 1 to 255 alphanumeric characters, semicolons and dashes are allowed. Spaces are not allowed.	1 to 255
Cache timeout (in minutes)	May be from 0-120 minutes.	1 to 3

ELEMENT	CONSTRAINTS	CHAR COUNT
User Name	Unique and case sensitive. The '@' character, underscores, dashes and spaces are allowed; however, no leading and trailing spaces are allowed.	1-80
User Password	Case sensitive. May be any keyboard character.	6-255
User Password for users assigned to SNMP Groups	Case sensitive, may be from 8 to 255 alphanumeric characters, and accepts the '@' character, underscores and dashes.	8 to 255
IP ACL Policy name	Unique & case sensitive. Accepts underscores and dashes.	1-32
User ACL Policy name	Unique & case sensitive. Accepts the '@' character, underscores and dashes.	1-32
Domain Policy Name	Unique and case sensitive. The '@' character, underscores, dashes and spaces are allowed.	1-80
Role Policy Name	Unique and case sensitive. The '@' character, underscores, dashes and spaces are allowed.	1-80