Siteminder Integration Guide
Integrating Siteminder with SA

Abstract
The Junos Pulse Secure Access (SA) platform supports the Netegrity Siteminder authentication and authorization server along with other common authentication servers. By using a single sign-on SMSESSION cookie, it also provides seamless access to all backend applications and servers protected by Siteminder authentication.

Overview
Along with authentication by Siteminder, SA also supports role mapping based on user attributes. In addition to LDAP and RADIUS, you can use Siteminder for retrieving user attributes that can be used in role mapping. When you authenticate SA users using a Netegrity Siteminder policy server, you can enable access to Siteminder protected resources without re-authenticating if authorized with the correct protection level. Additionally, you can re-authenticate users through the SA if they request resources above their current protection level. And you can enable users to sign into the policy server first and then access the SA without re-authenticating.

Operation
The Siteminder system has two major components:

- Web Agents, which run on each protected Web server
- Policy Server, which provides AAA functions within the Siteminder framework.

Siteminder is configured in a hierarchical fashion using Netegrity-specific entities: Policy Domains, Realms, Responses, Rules, and Policies. Within this framework, the SA is configured as a Siteminder Web Agent, protecting resources within the organization.

The SA uses a built-in Siteminder custom agent to authenticate SA users using Siteminder policy server configured for standard username and password, or ACE SecurID tokens, or Active Directory credentials, or client-side certificates for authentication. You can enable the agent to access Siteminder protected backend resources without re-authenticating. This feature allows single sign-on from the SA to Siteminder-protected resources using SMSESSION cookies.
There are two major steps to integrate Netegrity and SA:

- Configuring the SA Web Agent on the Siteminder Policy Server.
- Creating a Siteminder authentication and authorization server instance on the SA.

**Policy Server Configuration**

Before adding Siteminder as an SA authentication server, you must configure the SA as an agent on the Siteminder policy server. The following procedure outlines the general steps for adding an agent or agents to a Siteminder policy server. The images are captured from Siteminder policy server version 6.0.

1. To log into the Policy Server User Interface, click **Go to Start>Programs>Netegrity Policy Server User Interface**.
2. Click Administer Policy Server. A username and password is required to administer the policy server. This is the same username and password configured at the time of installation of policy server.

Default username is: Siteminder (Not case sensitive)
Password: none
Creating an Agent for the SA

3. Under System Configuration, right click on Agents and select Create Agent. Or from the Siteminder Administrator menu, select Edit > System > Configuration > Create Agent.

4. In the Name field, enter the name of the new Agent. This name is not case-sensitive, and must be 7-bit ASCII characters, in the range 32-127.

You cannot, for example, create one agent called "IVEAgent" and another called "iveagent".

**Note**: The value in the Name field must match the name entered in the AgentName parameter in the SA Configuration for Siteminder server.

   a. Name = win2k3  
   b. Web Agent is selected from drop-down list as the Agent Type  
   c. Check “Support 4.x agents”  
   d. IP Address = <internal port IP address>  
   e. Secret: Value specified here must match the value of the Secret parameter configured on SA Siteminder server.

5. To create a user directory, under System Configuration, right click on User Directories and select Create UserDirectory.
6. There are 5 name spaces available under directory setup:
   • LDAP
   • ODBC
   • WinNT
   • Custom
   • AD

Specify a unique name in list of user directories, then select the NameSpace as LDAP.

Specify the LDAP server’s (Win 2K3) IP address along with port number.

Under Root, specify the Base DN at LDAP Search.

To find user entries, configure LDAP User DN Lookup.

The Start field should have the value \texttt{CN=}

The End field has a Base DN with a comma in front. Policy server reads this as \texttt{CN=*}
under DN \texttt{cn=users,dc=intse,dc=nijuniper,dc=com}.
Click Apply and then click the Credentials and Connection tab.

The Credentials and Connection tab of the User Directory Dialog contains the Administrator Credentials box. If credentials are required for access, you must provide Siteminder with the appropriate credentials in the Administrator Credentials group box so that the Policy Server can access the user directory. Since the NameSpace is configured as LDAP, you need to provide the Admin DN of active directory along with password.

Click OK to save changes and finish user directory creation.

**Configuring a New Domain**

A domain is a logical grouping of resources associated with one or more user directories. In addition, domains require one or more administrator accounts that can make changes to the objects within the domain. Domains contain realms, rules, responses, and policies.
Example:

- Name = Win2k3Domain
- Add the user directories. You could add an existing directory from list of directories available or create a new directory using create button.

Click Apply to save the changes. When you configure a domain, you assign user directories and administrators and create realms.

**Note:** You can edit domain properties if you need to add an administrator or realm in the future.

Under the Domain dialog, click the Realms tab. Click Create to add a Realm under this domain. You could also create a realm and later add it to the domain by going to the Domains tab.

A realm is a set of resources with a common security and authentication requirement. Access to resources is controlled by rules associated with the realm that contains the resource.
Next to the *Agent field, click Lookup to select one of the available agents from the list.

Set the Resource Filter to /siteminder. When you configure the Siteminder authentication server settings in the SA Administrator Console, be sure to use corresponding settings for Protected Resource and Resource Action.

Select Authentication Scheme as Basic and Default Resource Protection as Protected (default).

Protected indicates that the resources in the new realm are protected from access by Siteminder when you save the new realm. Siteminder protects a resource as soon as it is placed in this realm. To allow access to this resource, you must create a rule and bind it to a Policy that allows access to users or groups.

**Note:** You can set the Realm Resource Filter to "/", and then all the resources passed to the policy server by the SA custom agent match the rule. By default, the Resource Filter is set to "/siteminder" in the SA Siteminder Server options, which is the recommended setting.

**Configuring a Session**
On the Session tab, you can set session timeouts, choose non-persistent or persistent sessions, and enable or disable synchronous auditing for the realm.

In the Session Timeouts Group Box, specify timeouts for each realm protected by a Siteminder Web Agent.

**Maximum Timeout Enabled** - If set, the values specified in the associated Hours and Minutes fields determine the maximum amount of time a user session can be active before the Agent challenges the user to re-authenticate. This setting is enabled by default. To specify no maximum session length, clear the checkbox. The default maximum session length is two hours.

**Note**: To use this feature with the Basic authentication scheme, the Web Agent must be configured to Require Cookies.

**Idle Timeout Enabled** - This setting is enabled by default. To specify no session idle timeout, clear the checkbox. The default session idle timeout is one hour.

The session actually expires after the specified idle timeout value. There is an extra time period given to the user which is determined by the number of seconds specified in the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\Siteminder\CurrentVersion\SessionServer\MaintenancePeriod
```

The default value is 60 seconds.

For example, if you set the idle timeout at 10 minutes, and you use the default value of the Maintenance Period registry setting, the longest time period before a session timeouts due to inactivity is 11 minutes (specified timeout + maintenance period).

**Note**: When Siteminder is used as authentication server to log into SA and session time out option is enabled, it overrides the SA session time out options. Then, only the session time out values configured on Siteminder server are considered. Also, the timeouts on policy server only take effect if both values are configured. If either one is not configured, then the SA timeout value is used. The timeout values are used from the realm authenticating the user. The timeout values in other realms, such as authorization, have no effect.

**Persistent Session Group Box**

In the Persistent Session Group Box, configure a realm to use non-persistent (standard) sessions or persistent sessions:

For persistent sessions, the Idle Timeout must be enabled and set to a value higher than that specified for the Validation Period.

**Creating Rules under a Realm:**

On the **Domains** tab, expand the Win2k3Domain and select Win2k3Realm. Right click and select **Create rule under realm**. Enter a name to reference the rule of this realm.

In the Realm and Resources Group box, select the realm and resources to apply the rule.

**Realm** - The realm with the resources to apply this rule. By default, the Realm is set to the name of the realm selected when you opened the Siteminder Realm dialog.

**Resource** - The Resource protected by this rule.

Below the Resource field, the Policy Server User Interface displays the effective resource protected by the rule. The effective resource consists of the Agent, all of the resource filters concatenated from any parent realms existing above the realm, the resource filter of the realm specified in the Realm field, and the resource you entered in the Resource field.

For example, if the following is created:
Realm1 Agent = Win2k3
Agent1 IP Address = 1.1.1.1
Realm1 with Resource Filter = /siteminder
Rule1 Resource = /myfile.html
Rule1 protects 1.1.1.1/<root directory for the Win2k3 Web server>/siteminder/myfile.html.

Perform Regular Expression Matching - If selected, specifies that a resource can be a specific file or an expression that uses resource matching with regular expressions.

**Action**
In the Action section, specify the type of event and the specific action that must take place for the rule to apply:

Web Agent Actions - Specifies that the rule applies to the HTTP action or actions selected from the list.

Authentication Events - Specifies that the rule applies for the authentication event selected from the list.

OnAuthAccept - Occurs when a user successfully authenticates. This can be used to redirect a user after a successful authentication.

In the Allow/Deny and Enable/Disable field, specify if Siteminder allows or denies access to a protected resource when the rule applies and if the rule is enabled.

The Advanced section contains two tabs:

Time Restrictions Tab - The Time Restrictions tab manages time restrictions for the rule.

Active Rule Tab - The Active Rule tab manages active rules for dynamic authorization based on external business logic.
Click OK to save changes and continue.

Right click on Win2k3Realm again and select Create rule under Realm to create a second rule. Name the rule with a new name. Under Action group box, select Authentication Events. Select OnAuthAccept. Click OK to save changes.
Creating a Response

Right click on Responses and select Create Response.
Click Create and Siteminder Response Attribute Editor is displayed.
Select WebAgent HTTP-Cookie Variable from the Attribute list.

Under Attribute Setup group box, select User Attribute. Under Attribute Fields, enter
*Cookie Name: nameofthecompany
*Attribute Name: company

**Note:** The Cookie Name is configured on the SA under role mapping using user attributes by specifying a value present in AD for attribute name.

Click OK to save changes and OK again to close Response.

**Creating a Policy**

Policies define how users interact with resources. When you create policies in the Policy Properties window, you link together objects that identify users, resources, and actions associated with the resources.

To create a policy:
From the Domains tab of the Policy Server User Interface, right-click a domain object and select Create Policy.
The Users tab binds users and groups to the policy. Add/Remove - Opens the Users/Group and allows you to specify the users and groups bound to the policy.

Click Add/Remove to add users or groups to the policy.
Under Manual Entry, select Search Users from the Action list. Under Available Members, click the Find icon to search for users. Select LDAP query and enter cn=*.

Click OK.
The Active directory users and groups are displayed in Available Members window. Select required users/groups to apply this rule and move them to Current Members. Click OK to return to the main policy dialog window.
The Rules tab allows you to specify rules and rule groups to include with the policy. Navigate to the Rules tab and click Add/Remove. The previously created rules are displayed. Select rules from Available Members and move them to Current Members.
Click OK to continue. Select Win2k3RealmRule1 and click SetResponse.
Select Win2k3Response you created and click OK.
The IP addresses tab allows you to specify an IP address, a range of IP addresses, a subnet mask, or a host name for use in policy and associate objects.

The Time Dialog is used to configure time restrictions for rules, policies and associates. A time restriction indicates specific time periods during which a rule or policy applies or when users access an associate.

Creating Siteminder as an Authentication and Authorization Server on the SA

Login as administrator, and navigate to Authentication/Authorization servers.

1. Specify a name to identify the server instance. You can only create one Siteminder authentication server.

2. Enter the name or IP address of the Netegrity policy server. You can also enter a comma-delimited list of backup policy servers. If you enter one or more backup servers, choose a Failover Mode. Select Yes to have the SA use the main policy server unless it fails, or select No to allow the SA to load balance among all the specified policy servers.
3. Specify the shared Secret and Agent Name configured on the policy server. These must match EXACTLY the configuration on the Policy Server, and only Secret is case sensitive.

4. If you do not want users to see the SA sign-in page, specify a URL to redirect users when they sign out of the SA.

5. Under Protected resource and action:
   a. This setting must match that configured on the Policy Server. It is recommended that you accept the default setting of /siteminder in the Protected Resource field. When a user signs in, the user is authenticated against the Siteminder Server with this administrator specified protected resource.

   b. Specify the Resource Action that corresponds to the action, such as GET, POST, or PUT, set for the policy server. Typically you can just enter GET in this field, as long as GET was one of the Web Agent Actions defined on the rule for the protected resource defined above. Since the SMSESSION cookie is passed back to the browser as a session cookie, you need to define which domain(s) to transmit the cookie from the browser.

Configuring SMSESSION Cookie Settings

a. In Cookie Domain, enter the valid Internet domain for the cookie and where the browser of the user sends cookie contents. This cookie domain should be the same as the SA domain. (example: .jnpr.net)

   b. In the Cookie Provider Domain field, enter the valid Internet domain of the cookie and the location where the SA sends cookie contents. If you have configured a cookie provider, enter the domain of the cookie provider. If not, enter the domain of the Web agents for the desired single sign on requirement. This setting can be used to apply single sign on to remote Websites by defining those remote domains as a comma separated list in this field. (example: .jnpr.net, .netscreen.com)

   c. If other Web agents are set up to accept secure cookies, choose Yes for Send cookie securely? The cookie is then communicated only through HTTPS. To send a cookie over HTTP, choose No.
Note: If Send cookie securely is set to HTTP, Siteminder sends the cookie over both HTTP and HTTPS. If set to HTTPS, then it is only sent over HTTPS.

6. Under Siteminder Authentication, perform the following steps:
   a. To automatically sign in users who have a valid SMSESSION cookie, select Automatic Sign-In, and then select the authentication realm. Use this option to provide a single sign-on experience for users. This option enables users to sign in using a standard Siteminder Web Agent that generates an SMSESSION cookie. When a user clicks on a link to the SA (hostname), this cookie is passed to the SA and the user is not prompted to submit their credentials again. All users who sign in to the SA this way are mapped to the realm or role that you specify, and any access controls specified for the role apply to it.

If you enable the Automatic Sign-In option as shown below, the SA can use SMSESSION cookie which is already present in the browser to enable single sign-on to the SA.

   The SA does not support the Automatic Sign in feature for administrator roles. This feature is only available for users.

   b. When users sign in directly to the SA, there are two methods by which they can obtain an SMSESSION cookie. You must select one of the two methods. The choice of authentication method is independent of whether you select to allow Automatic Sign In above.

   - **Authenticate using custom agent**—Select this option if you want to authenticate using the SA. Using this option, the SA generates the SMSESSION cookie, just like any other Web Agent configured within the organization. Currently, using this option requires that you update all other Web Agents to accept the IVEgenerated cookie, and apply a software patch to all other Web Agents. For this reason, you may want to choose the other method.
• **Authenticate using HTML form post**—Select this option if you want to authenticate using another WebAgent. Instead of SA directly contacting Siteminder with the user-entered credentials, you post the credentials to another previously configured standard WebAgent. When a user accesses a resource on a particular Web server, the WebAgent on that Web server contacts the Policy Server and then loads the user login page. The user enters their credentials and, if validated, the user accesses the available resources. In this case, Siteminder acts as a browser and posts the credentials to the standard Web Agent.

![Diagram of Siteminder Integration](image)

In this example, the Target, protected.jnpr.net, is an internal Web server protected by the same policy server as the SA. In this case, the SA acts as a custom agent and the agent deployed on the internal Web server is a standard agent. The resource, protected.jnpr.net/resource, is protected by the policy server. When a user tries to login to the SA, SA accesses /siteminderagent/forms/login.fcc on the internal Web server where the Web agent is deployed and sends the URL http://protected.jnpr.net/siteminderagent/forms/login.fcc/?target=http://protected.jnpr.net/resource. The target is the URL that the standard agent redirects the user, after successful authentication.

![Diagram of Siteminder Integration](image)

When you contact JTAC with issues about authenticating using HTML FORM POST, JTAC needs the TCP dump logs on SA as well as the TCP dump logs on the client PC when it accesses the Siteminder Web agent directly(without SA).

7. Click **Save Changes**. The server instance is saved and appears on the **Servers** page.
Role mapping

Unlike an LDAP server instance, however, the Siteminder server instance name does not appear in the Directory/Attribute list of a realm. To use a Siteminder server to retrieve user information, you simply select the instance name in the Authentication list and then select Same as Above from the Directory/Attribute list. Then, you configure role mapping rules to use attributes from the Siteminder server, which the SA provides in an attribute list on the Role Mapping Rule page after you select Rule based on User attribute.

Navigate to the role mapping section and create a new rule based on user attribute.

From the Rule Based on list, select User attribute and click Update.

Click Attributes to get the Siteminder server catalog as below. Enter name of the attribute and click Add Attribute. Click OK to save changes. Note that Name of the attribute is the cookie name specified in the Response attribute of the policy server.
Specify the value of the attribute and map the user attribute value that matches the one configured.
Siteminder configuration along with role mapping based on one of the user attributes from AD is complete.

**Note:** While logging in as a user, make sure you access SA using a FQDN, and not the IP address. The FQDN should have the domain name provided in SA Siteminder configuration SMSESSION cookie settings.

If you attempt to log into SA using its IP address or short name instead of FQDN, you may see the following entry in user access logs, “The admin configured Cookie Domain, on the Siteminder Authentication Server, does not match the SA domain for user <USERNAME>/<User Authentication Realm>.”

This indicates that after successful authentication to the policy server, login is rejected because the domain name in the SA user access URL does not match with the one configured on the SA Siteminder server.

You cannot define Netegrity Siteminder server as a secondary authentication server. Also, if a Netegrity Siteminder server is defined as the primary authentication server, you cannot define a secondary authentication server for that realm.

**SMSESSION Cookie**

A **SMSESSION cookie** is a security token that stores user credentials and protection levels. Depending on configuration, the Siteminder Web agent (SA) creates a SMSESSION cookie to store the sign-in data, and then posts the SMSESSION
cookie to the following locations so that the user does not have to re-authenticate if he wants to access additional resources protected by Siteminder. If the user tries to access a Siteminder resource from within the SA session, the SA sends the cached SMSESSION cookie to the Web agent for authentication.

**Support for Handheld devices and PDAs**

Since Siteminder depends on cookies, and most i-mode browsers do not support HTTP cookies, the SA supports all of the other authentication servers except the Netegrity Siteminder policy server.

**Debugging**

Look for following information when trying to debug Netegrity Siteminder issues.

1. **SA Logs:** The User Access log provides every failed attempt for authentication/authorization issue. There are log entries if authentication request fails, authorizing request fails, validation of cookie fails or key rollover, etc. If there is any issue with role mapping based on user attributes, the policy trace log on SA with Role mapping events logged should help.

2. Check the Policy Server Authentication and Authorization log files. Policy Server logging is configured using the Policy Server Management Console. In the Logfile field, specify the path of the log file. In the Policy Server Audit Log, select the events to log Authentication and Authorization attempts. The options are
   - Log No events
   - Log Rejection events only
   - Log All Events

**Common Debugging Problems**

1. Make sure the SA system time is synchronized with Siteminder server. If not, timeouts may not function correctly and users may not be able to login.
2. Make sure the proper Session Timeouts, including maximum timeout and idle, are defined in Siteminder Realm Dialog.
3. After you sign into SA and browse to a Netegrity-protected Webagent, you see the Netegrity login page instead of SSO. Verify that the Cookie Provider Domain matches the domain of the netegrity-protected Webagent. Verify that Send cookie securely is selected. If “yes”, then SSO works only with https:// site. If “no”, SSO works with both http:// and https:// sites.
4. For SSO failures from the SA to a Std agent, if the std agent log contains messages like the following:

   - [06/Jul/2005:11:24:52-1504-42-0] EstablishSession - **ERROR** **cookies are non-transferable.**
     - CookieUser='CN=Test_User,OU=Support,OU=USA,DC=acmegizmo,DC=com' CookieIP='10.10.10.10', User IP passing cookie = '2.2.2.2'
   - [06/Jul/2005:11:24:52-1504-42-0] EstablishSession - Exiting with HTTP 500 Server Error: 00-0003
This appears to occur in environments where a proxy is configured. This may be due to the PersistentIPCheck or TransientIPCheck parameter being enabled on the standard agent.

**Enabling IP Checking**

To prevent a breach of security by an unauthorized system, you can enable or disable IP checking with persistent and non-persistent cookies. An unauthorized system can monitor packets, then steal a cookie, and use that cookie to gain access to another system. The IP Checking feature enables the Web Agent to compare the IP address stored in a cookie from the last request with the IP address in the current request to see if they match. If they do not match, the Web Agent rejects the request.

To enable this feature, add the persistentipcheck and transientipcheck values to the registry as follows:

1. Select Start | Run.
2. In the Open field, enter regedit and press ENTER.
3. Navigate to HKEY_LOCAL_MACHINE | SOFTWARE | Netegrity | Siteminder Web Agent | Microsoft IIS
4. Select Edit | New | DWORD Value
5. Double-click on the new value and configure the fields as follows:
   a. Value name:
      - persistentipcheck (for persistent cookies)
      - transientipcheck (for non-persistent cookies)
   b. Value data: 1