Introduction

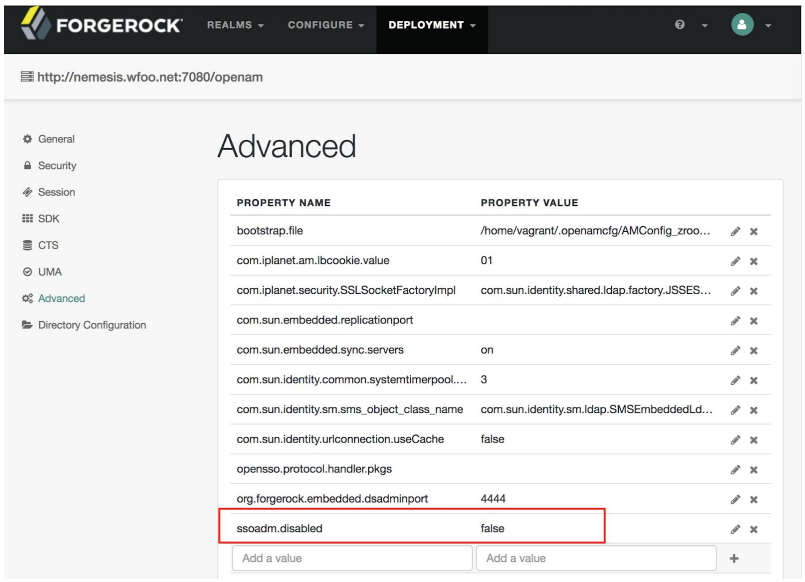
This Guide provides step by step instructions on installation and configuration of the ForgeRock SiteMinder Migration Toolkit. The toolkit comprises of different components such as scripts and plugins to facilitate coexistence and migration form a SiteMinder environment to a ForgeRock environment.

Installation and Configuration

Authentication Plugin for SiteMinder

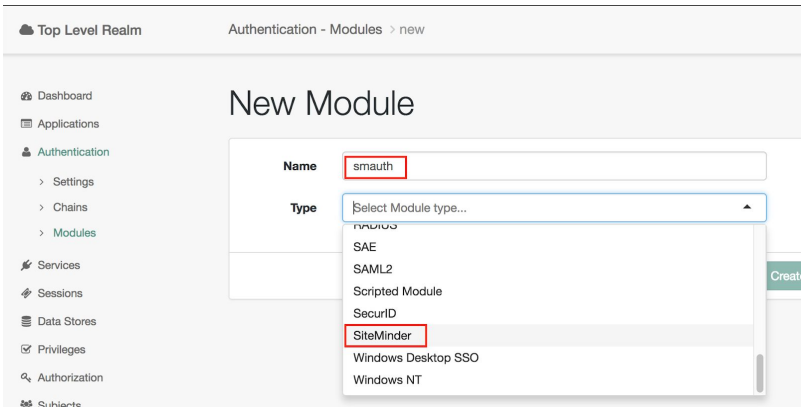
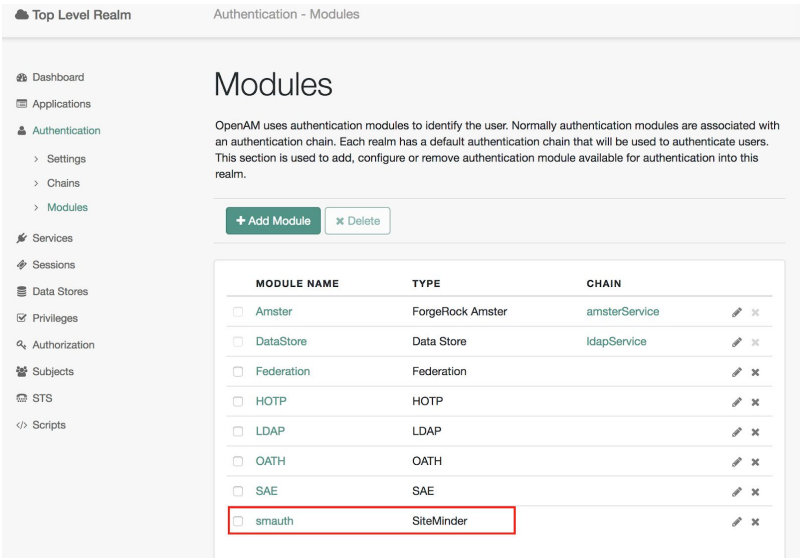
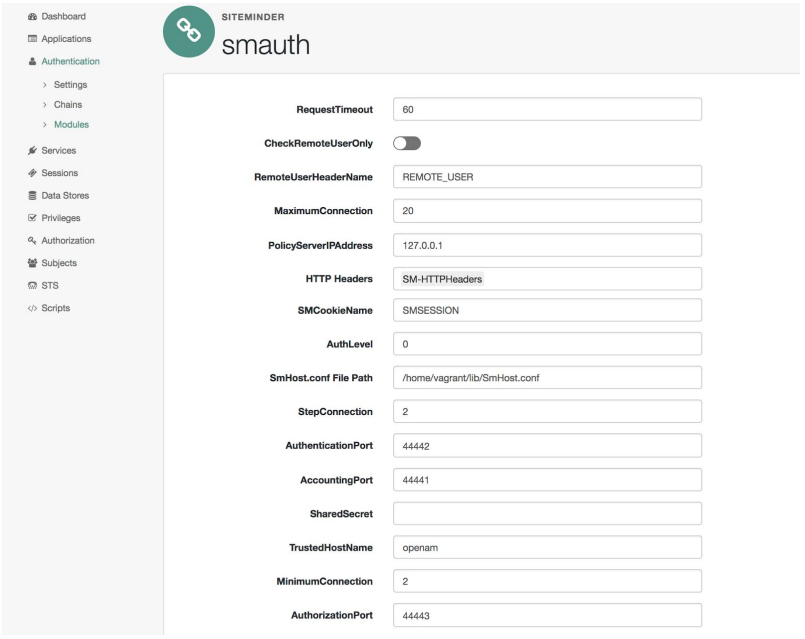
1.0	First you need to build a custom war file to include the libraries for the plugin.
	
	This step adds the required openam-auth-siteminder-x.x.x.jar file to the WEB-INF/lib directory as well as the associated resource and properties files to the relevant directories. Note en locale is assumed. For any other locale you will have to manually add the corresponding property files.
	
	As the this plugin uses the SiteMinder API you also need to add the cryptoj.jar and smagentapi.jar files manually to the WEB-INF/lib directory. These files can be obtained from the SM SDK distribution.
		$ mkdir /tmp/amwar
		$ unzip openam.war -d /tmp/amwar
		$ unzip openam-auth-siteminder-x.x.x.zip
		$ cd openam-auth-siteminder
		$ ./install.sh /tmp/amwar
		
		⇨ Manually add the *cryptoj.jar* and *smagentapi.jar* files to the WEB-INF/lib directory of the exploded openam.war file
		
		$ cd /tmp/amwar
		$ jar cvf /tmp/openam.war .
		
		⇨ Now deploy this customized openam.war file to your container and then configure openam as usual.
		
		Note: In a dev environment on Tomcat you can install the plugin directly by running
		$ ./install.sh /home/wahmed/tomcat/webapps/openam
		
		⇨ Restart tomcat
		
		Note: Ensure you are using the pure java API (smagentapi.jar) and not the legacy JNI linked with C libraries (smjavaagentapi.jar). If you use the legacy jar file then make sure that the LD_LIBRARY_PATH is correctly set so that the correct libsmjavaagentapi.so is found by the loader.

1.1	For this AuthN Plugin to communicate to the SM Policy Server, ensure that you register a trusted host and generate the corresponding hosts file as well. The smreghost binary or smhostreg.sh script are part of the SDK distribution.	$ ./smreghost.sh -i 192.168.50.9 -hn openam -hc HCO -u siteminder -p Admin123 -f /tmp/SmHost.conf Note: The SmHost.conf file will be used in subsequent steps to configure the AuthN module.
1.2	There are two ways to register a custom auth module in OpenAM. Either via the ssoadm command line tool or via the UI using ssoadm.jsp. For reference see the chapter in the Developer's Guide on the Sample Auth Module.	If you decide to use the ssoadm.jsp, first you would need to enable it by adding to the Advanced properties of the Server as shown below. A restart is recommended.

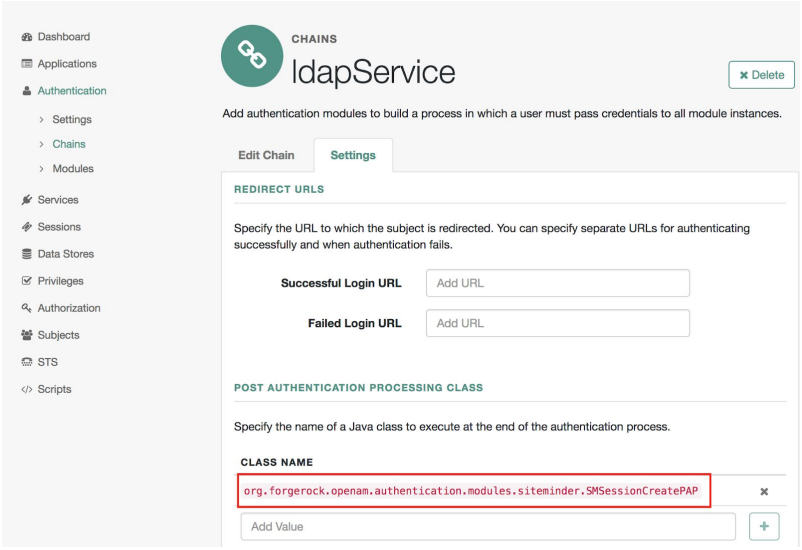
1.3	Register the SiteMinder Authentication Plugin by creating a service either by using the ssoadm.jsp method or the ssoadm command line utility.	**Using ssoadm.jsp**
		➪ Create the service by navigating to http://am.wfoo.net:7080/openam/ssoadm.jsp?cmd=create-svc
		➪ Paste the full contents of *SMAuthService.xml* in the textbox and Submit
		➪ Restart the container.
		➪ Register the new Auth Module by navigating to http://am.wfoo.net:7080/openam/ssoadm.jsp?cmd=register-auth-module
		➪ Paste in the textbox `org.forgerock.openam.authentication.modules.siteminder.SMAuthModule` and Submit
		➪ Check the debug/CoreSystem log file to make sure that there is no error in particular if you see the following
		ERROR: AuthenticationModuleCollectionHandler.handleQuery(): Invalid auth module instance configuration: SMAuthModule.
		Restart the container to resolve this problem.
		
		Using ssoadm command line
		`$ ssoadm \`
		`create-svc \`
		`--adminid amadmin \`
		`--password-file /tmp/pwd.txt \`
		`--xmlfile src/main/resources/SMAuthService.xml`
		
		`$ ssoadm \`
		`register-auth-module \`
		`--adminid amadmin \`
		`--password-file /tmp/pwd.txt \`
		`--authmodule org.forgerock.openam.authentication.modules.siteminder.SMAuthModule`

1.4	Now create an instance of this authentication module by navigating to the “Authentication” section, clicking on “Modules” and adding the	

	"SiteMinder" module from the dropdown. Call this module "smauth". á If you don't see the SiteMinder module in the dropdown then you failed the register the module in steps 1.2. Go to the troubleshooting section of this guide for help.	 
1.5	Click on smauth and configure it to match your environment. If you are using the *SmHost.conf* file then you don't need to supply the SharedSecret or the TrustedHostName. Note that the SmHost.conf file was generated in step 1.2.	
1.6	Test this auth module. Remember this auth module will create an OpenAM session if a SiteMinder session exists i.e. a valid SMSESSION cookie exists in the request.	First log into SiteMinder Then point the browser to http://am.wfoo.net:7080/openam/XUI/?module=smauth The next thing you see is the user's profile page

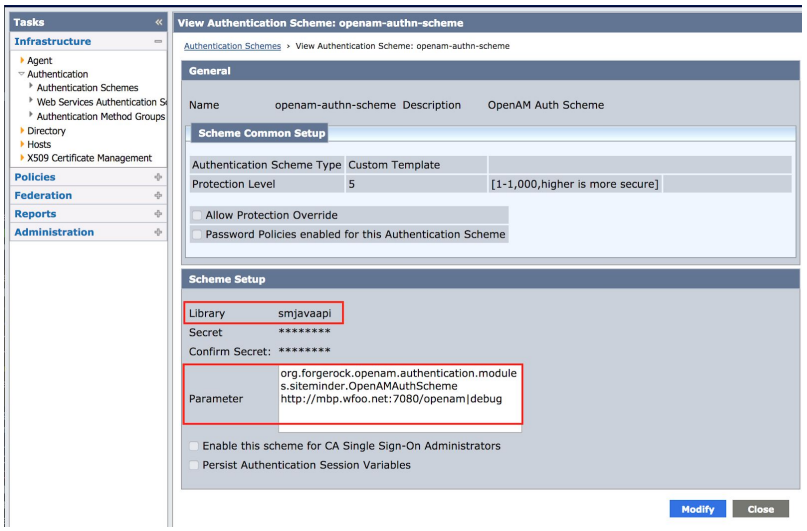
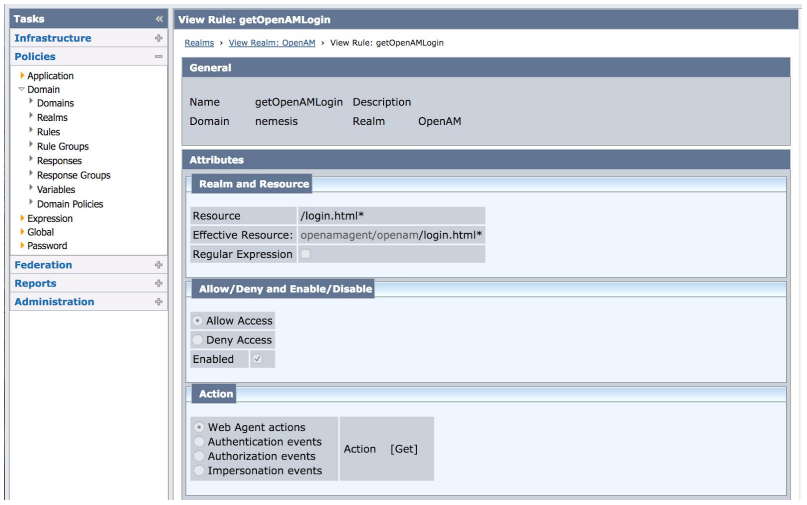
		If you get an error then goto the troubleshooting section of this guide for help.

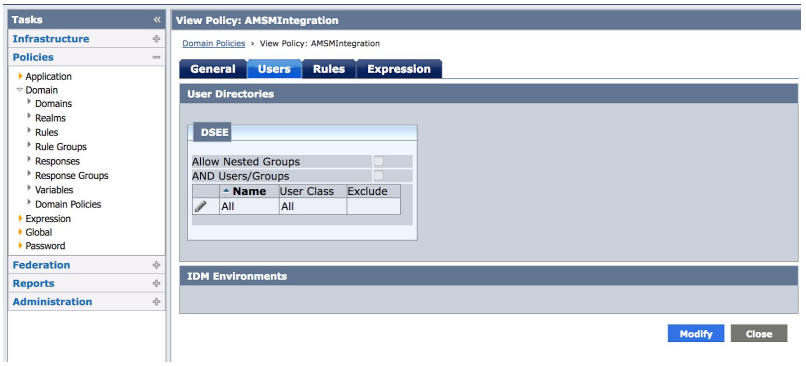
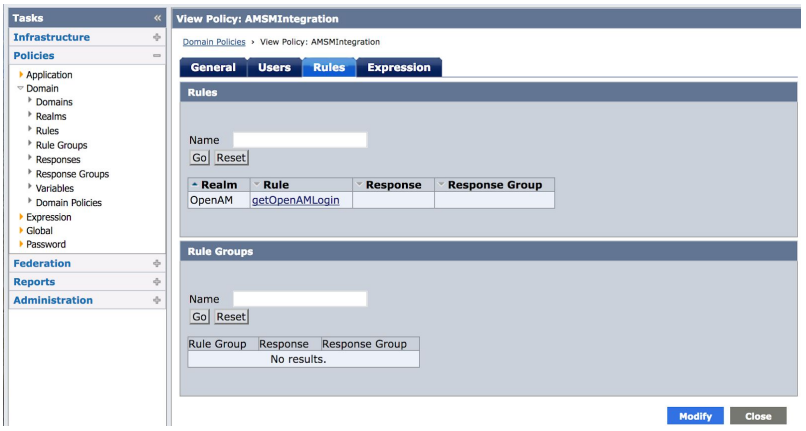
Post Authentication Plugin for SiteMinder

2.0	Add the optional Post Authentication Plugin (PAP) to allow creation of SiteMinder session simultaneously when a user logs into OpenAM. The PAP requires the creation and configuration of a Custom Authentication Scheme on the SM Policy Server using the adminui. The process is described in the next row.	⇨ Navigate to http://nemesis.wfoo.net:7080/openam/XUI/#realms/%2F/authentication-chains/edit/ldapService ⇨ Add the following class to "POST AUTHENTICATION PROCESSING CLASS" for your default Chain which in this case is ldapService. `org.forgerock.openam.authentication.modules.siteminder.SMSessionCreatePAP` 
2.1	Add the corresponding SM Custom Auth Scheme. These are the openam-auth-siteminder-x.x.x.jar and the json-xxxxxxxx.jar file.	⇨ Append to "java.class.path" siteminder/config/JVMOptions.txt the AM Client SDK *resources* directory path and all the jar files under lib as follows `-Djava.class.path=/zroot/ca/siteminder/resources:/zroot/ca/siteminder/config/properties:/zroot/ca/siteminder/bin/jars/smbootstrap.jar:/zroot/ca/siteminder/bin/jars/openam-auth-siteminder-1.0.0-SNAPSHOT.jar:/zroot/ca/siteminder/bin/jars/json-20160810.jar` $ ./stop-ps && ./start-ps

Custom Authentication Scheme on SiteMinder

As a prerequisite you need to have completed step 2.1.

3.0	Log into the CA Policy Server Admin UI and perform the following tasks: Create a custom authentication scheme as shown on the right. Note the “debug” parameter should be removed in production.	
3.1	Create a new realm and assign this authentication scheme to it. The realm can be created in an existing Domain or added to a new Domain. In this new realm create a new rule as shown. Note it is recommended that you use a dedicated webserver instance and agent for this integration. However you can use an existing one also keeping in mind that the protected resource (path) is created in either case as illustrated in step 3.5.	
3.2	Add a new Rule	

3.3	Create a Domain Policy. Make sure you configure the same user repository as the one OpenAM is pointing to or that the two disparate repositories are synchronized.	 
3.5	Create <apache-docroot>/openam/login.html	
3.6	Add the openam-auth-siteminder-x.x.x.jar file and json-xxxxxxxx.jar (from json.org) to the policy server's “java.class.path” in config/JVMOptions.txt	-Djava.class.path=/zroot/ca/siteminder/resources:/zroot/ca/siteminder/config/properties:/zroot/ca/siteminder/bin/jars/smbootstrap.jar:/zroot/ca/siteminder/bin/jars/openam-auth-siteminder-1.0.0-SNAPSHOT.jar:/zroot/ca/siteminder/bin/jars/json-20160810.jar

Token Transformation Plugin

4.0	Not Available Yet	
4.1		

Troubleshooting

Area	Error	Resolution
Smps log	[SmJavaAPI.cpp:639][ERROR][sm-JavaApi-00670] SmJavaAPI: Unable to get a JVM environment	There are a few reasons for this error but in this particular case it was due to incorrect JDK path setting to just “/usr”. Hence smjavaapi was not able to find the appropriate *libjvm.so*. Ensure that in *ca_ps_env.ksh* proper path is defined. For example `NETE_JDK_ROOT="/zroot/orcl/jdk1.8.0_131"`

		NETE_JRE_ROOT="/zroot/orcl/jdk1.8.0_131/jre"
Smps log	[SmAuthServer.cpp:339][ERROR][sm-Server-02940] Failed to query authentication scheme 'openam-authn-scheme'	If your authentication scheme is defined in the wrong domain or is not associated with the correct realm, you will see this error.
Agent log	[sm-AgentFramework-00520] LLA: SiteMinder Agent Api function failed - 'Sm_AgentApi_IsProtectedEx' returned '-1'.	If this is the last line that you see in the smps.log when the custom auth scheme debugging is enabled, it means that the openam client sdk is missing libraries from the classpath in JVMOptions.txt.
SiteMinder debug log file of openam	SiteMinder:04/25/2017 12:01:29:596 PM EDT: Thread[http-nio-7080-exec-9,5,main]: TransactionId[055371c0-3cf6-4f40-a3c6-a3bae6e5a554-155] ERROR: SMSessionUtils.createSmSession() Siteminder authentication unsuccesful, user=wahmed, Set-Cookie=[SMCHALLENGE=; expires=Thu, 27 Oct 2016 16:01:29 GMT; path=/; domain=.wfoo.net]	
SiteMinder debug log file of openam	TransactionId[e94de590-c5ca-4d10-9cc6-54dc379da7b0-75095] ERROR: SMSessionUtils.createSmSession() Siteminder authentication unsuccesful, user=wahmed, response=404	Ensure that you have the login.html file in the web server's docroot/openam/login.html directory or any other path of your choice to which the SiteMinder Policy Realm is pointing to.
CoreSystem debug log file of openam	Caused by: java.lang.NoClassDefFoundError: Could not initialize class com.ca.siteminder.sdk.agentapi.connection.SmAgentApiConnection	First take a look at https://docops.ca.com/ca-single-sign-on/12-52-sp1/en/programming/programming-in-java/agent-api-guidance-for-java#AgentAPIGuidanceforJava-ImplementthePureJavaAgentAPI

REFERENCE

- https://wikis.forgerock.org/confluence/display/openam/Write+a+custom+authentication+module#Writeacustomauthenticationmodule-RegisterthemoduleandtheservicewithOpenAM