GFIPM Phase I – Lessons Learned

1 Executive Summary

This purpose of this document is to share the knowledge gained during the Private Defense Attorney (PDA) project.

We have three members in our development team each were assigned different roles that worked successfully for the Global Federated Identity and Privilege Management (GFIPM) project. Member A was assigned the R&D role and was in charge setting up and configuring the environment. This included the following steps:

- Setting up an Oracle 11g database to be used with the Oracle Entitlement Server (OES) Product.
- Creating Groups and Policies within OES.
- Installation and configuration of a WebLogic 11g server for use with OES and Oracle Identity Federation (OIF).
- Installation and configuration OIF.
- Creation of custom WebLogic plug-ins such as OES functions and attribute retrievers.
- Incorporating the OES Java API into an existing production application that is running in a development environment. Hard coded policies have been replaced with flexible policies that have been externalized and reside in OES.

Once each major piece of the puzzle was figured out, the information was passed to Member B for support when the product gets moved into production. Member B was assigned a support role which included working with the Member C to prepare installation and configuration documents. Member C was assigned the task of Technical Writer. Every facet of installation & configuration was documented with clear instructions and a multitude of screen shots.

2 Project Overview

GFIPM has two federated frameworks to choose from: Security Assertion Markup Language version 2.0 (SAML 2.0) and WS-Federation. We chose to go with SAML 2.0 because it was a more straight forward framework to understand and implement. Security was a primary concern for the team when deciding to use SAML 2.0 for our federated framework. WebLogic 11g provides an excellent security framework for application protection. Using SAML 2.0, an application can be deployed without requiring the Service Provider (SP) to host a local logon screen. Users are unable to access the application unless they log in through an Identity provider (IdP). In the past, our application developers were required to create a home grown security screen that would keep unauthorized users out. By removing this step in the application development process, we increase security and decrease development time.

WebLogic 11g is an ideal solution because it includes support for a SAML 2.0 based Service Provider. Applications hosted on WebLogic can be federated relatively easy using built-in URL's. For example, to access our application, a trusted partner would access the following link: http://${sp.host}:${sp.port}/saml2/sp/sso/initiator?IdpName=IdP_name&RequestURL=/targetapp/target.jsp Using these links, we can create a central portal with a custom Discovery Service that would allow our partners to choose their Identity Provider from a drop-down list. Once selected, it would redirect to their Identity Provider for authentication.

Using SAML 2.0 in WebLogic 11g provides the authentication for our applications; however, we needed to provide fine-grained authorization using attributes passed in the SAML assertion to create policies on these attributes.

We initially researched a product called Oracle Access Manager (OAM) for our fine-grained authorization solution. OAM can integrate with OIF. However, the only way to use OAM with a fine-grained mechanism was to use the SAML Attribute Sharing profile. After contacting Georgia Technology Research Institute (GTRI) we found out that the SAML Attribute Sharing was not an accepted profile for the GFIPM grant statement of work. Due to this our team had to find another product that would satisfy the fine-grained authorization requirement. Our team was pleased to search for another product. We found the steps to integrate OIF with OAM were too involved and complicated. Many low level configurations were necessary to make them communicate. In addition, we found it difficult to create policies using the Access Manager console. We were provided one week of training on how to use OAM but the class only confirmed how difficult the product is to setup and use.

Oracle Entitlement Server's framework is specifically designed to provide fine grained authorization based on Groups and Attributes. By the policies being externalized outside the application, the policies can be changed dynamically by a security administrator. As business requirements are constantly in flux and more often than not, the application developer needs to change the code and redeploy the application, but with Oracle Entitlement Server, the change can be done in a matter of minutes without an application developer's involvement.

Our greatest challenge was integrating OIF with OES. One of the biggest issues in the initial portion of phase 1 was that the meetings with Oracle Consultants and experts never told us that SAML 2.0 and OES could not be integrated. Ultimately, our team had to change Oracle WebLogic code by reverse-engineering a core Jar file to get the project working. This may or may not be acceptable for a Production environment; however, using the products "Out of the Box" would never work without this custom hack. That said, the Oracle experts should have told us that SAML 2.0 and Oracle Entitlement Server cannot work together.

Later in the R&D phase of the project, Oracle provided a great consultant to assist in solving our OES and OIF integration. He provided information on both OES and OIF that were instrumental in solving our integration problem. Please see assisted tasks below:

- Helped with OES configuration and setup
- Assisted in steps needed to pass attributes from OIF
- Provided solution on how to handle multiple SP (Service Providers) and Identity Providers using built-in WebLogic URL strings.
- Provided a working functional example for OES.
- Answered many questions regarding OES and OIF

For the Identity Provider (IdP) we needed to store user identities in an Lightweight Directory Access Protocol (LDAP) repository. Our team chose a product called Oracle Internet Directory (OID). Our goal was to use the OID 11g because it is the most current version and is supposed to integrate with the centrally managed Fusion Middleware control. We spent approximately a month trying to install the product with no success. We tried installing the OID 11g on Windows 2003 R2 and SUSE Linux. Regardless of what options or work-arounds we tried, the product would always fail near the end of the installation process. We even created a Technical Assistance Request (TAR) with Oracle but they were unable to provide any answers as to why the installation failed. We suspected the cause was related to running a separate configuration process called Repository Creation Utility (RCU). This Oracle utility inserts the OID 11g schema objects into an Oracle database. We noticed on various technical blogs that other users were experiencing the same problem and some had success after deleting their database and rerunning the RCU. We had no such luck with our installation. Since many of the users are posting SOS blogs on the Internet, we suspect within a couple of months if we download the current version of OID 11g, the install will most likely work. Ultimately, we had to use OID 10g for our LDAP repository.

OID 10g is very stable and has been around a long time. The installation went smoothly with no problems.

Our team attended a one week class in Oracle Directory Services 11g (ODS) which includes OID and Oracle Virtual Directory (OVD). We believe that OVD can play a critical role in our identity management infrastructure. One key feature of OVD is that database tables can be virtualized to work as an LDAP schema object. Currently, we have an application that authenticates against a relational database. A possible strategy could be to virtualize the user database table so that it appears as an LDAP schema object. This way, we can create a consistent LDAP structure that merges OID with database users. There would only be a single LDAP system to connect to making authentication easier to setup. Another advantage would be that it would allow us to virtualize tables that contain attributes that relate to the user table. Attributes that exist in the database could be combined with attributes in OID to create a seamless view.

Oracle Identity Federation 11g is a very robust product with many powerful features that should suffice for most federated environments. Setting up an IdP and SP were a straight forward process. Our development team appreciated that it was integrated in the Fusion Middleware for central management.

We initially had problems due to GFIPM attributes being included in a single base64 attribute that needed to be extracted manually. Oracle Identity Federation only supports flat attributes for direct mapping. Fortunately, GTRI created a new flat attribute standard that we were able to use.

The first phase of the project that we worked on is called Public Defense Attorney (PDA). The PDA project is currently in our production environment but it is not using SAML 2.0 or fine-grained authorization. Our team removed the traditional logon screen and replaced the security with SAML 2.0. In addition, all the hard coded private attorney business rules were replaced with flexible policies that can be configured in OES. For example, when an attorney is has an account on our system, we assign them an expiration date one year in the future. Fifteen days before their account is going to expire, we give them a warning message that their account is going to expire. The fifteen day value was hard coded into the application and the only way to change it was to assign a programmer to change the value and re-compile and re-deploy the application. With the new setup, a policy reference was placed in the code section using the OES Java API where the fifteen day check was performed. The policy reference is essentially a Boolean check based on a policy name that exists in OES. Since the business policy is now in OES, the fifteen days can be easily changed to twenty days without recompiling the code. Please see the OES policy example below:

- Policy Name - UI/AccountExpirationAdvanceNoticePolicy
  - Actual Policy - Warn_of_future_expiration_date(SecurityClearanceExpirationDate,-15) and EmployeePositionName = “Private Attorney”

A custom function was created as a plug-in to OES to perform a present day or future date check. By changing the -15 parameter to -20, the policy will check twenty days in advance to warn the user that their account will expire. Also, the second part of the policy with the EmployeePositionName = “Private Attorney” check will evaluate if the person using the system is a Private Attorney. A message policy was setup that works in conjunction with the above policy. The message resource is shown below:

- AppMessages/AttorneyFutureExpMsg
  - Report_as(“AttorneyFutureExpMsg”,”Warning! Your subscription will expire on $1”) and report(SecurityClearanceExpirationDate)

In the application, there is a policy reference that is put into the Java code to replace the hard coded message. Using the OES Java API, the report_as function is used to retrieve text values from OES. This works great for creating custom messages to display to the user. The second part of the policy is the report function. This allows values from any user attribute to be retrieved by OES from the Java application. Since OES does not have a built-in place holder replacement feature, we created are own way of creating dynamic messages based on user attributes. After the two parts of the policy are

retrieved into the java application (“Warning! Your subscription will expire on $1” and report (SecurityClearanceExpirationDate) (e.g. 04/03/2010), they are combined to create a complete message: “Warning! Your subscription will expire on 04/03/2010.” There were a few more policies that we implemented for the PDA project and they can be found in the PDA demo document.

Our development teamed used VMWare for both the Oracle Identity Federation server and Oracle Entitlement server. On many occasions, the Oracle environments would get corrupted from changes made. Using VMWare, we were able to restore a snapshot of the working model before changes were made. This made our R&D process more efficient since we were not concerned that a configuration change would corrupt the project.

Overall, the Oracle suite was able to satisfy the GFIPM grant providing all the functionality that the grant dictated. Even though the project was completed successfully, there were many issues that were difficult to overcome. We listed the challenges below in each paragraph.

3 Project Issues

OES times out in about 60 seconds if not in use. This feature is annoying and makes it difficult to use. The timeout should be longer or configurable. Also, a checkbox in OES needs to be checked each time you log in or you will lose your work if you made any changes without saving. The combination of the system timing out in about 60 seconds and the box not being checked by default is setting up a user for a lot of frustration when they lose their changes.

Oracle installation for OES is too complex and involved. It requires low level configuration and the running of various batch files to configure. In addition, various cp2 patches need to be applied or the application will not work. This is the first product our team has ever installed that would not work without a patch. To get the patch, we had to find it in Metalink. Fortunately, Superior Court has a Metalink account and we were able to obtain the patch. The patch should be included on the same OES product download page. This would help ensure that the patch is not missed by mistake.

Our team was unable to get the policy simulator working with OES. We installed the WebLogic Security Services Modules (WLS SSM) but were unable to run policy simulations within OES. This feature would be very valuable when creating policies.

When using the ASI screen in, a Java Applet plug-in will not work. We have tried to match the exact Java version to the one the OES is expecting without successful results. We can work around this issue by navigating on the right pane; however, it gives the user the impression that the product is not stable and that Quality Assurance testing was neglected.

When setting up and running SAML on the WebLogic server, we received an error that “<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">” is not allowed. WebLogic would not accept an Entity format. This was is an Oracle bug. Fortunately we had a contact in Oracle and they provided a patch jar file that fixed the problem. We were lucky to get the patch because the patch jar file was not available to the public at this time.

When using OIF, we attempted to use colons “:” in the attribute names. The system gave an error message stating the colons “:” are not allowed. After attempting this change and restarting OIF, the Federation SP disappeared. When trying to load it back, the system would state that it already exists. Fortunately, we knew where the underlying configuration files were located. When we opened the cot.xml file, the colons “:” were entered into the system even though the validation appeared to block the entry. This caused hours of troubleshooting to solve. Oracle needs to fix this problem. The colon issue surfaced because of a GFIPM requirement to have colons in the name. For example, one of the required attribute names is gfipm:2.0:user:FederationId. This should be an allowed attribute name when using urn:oasis:names:tc:SAML:2.0:attrname-format-uri.

The accepted name ID formats in GFIPM are Transient & Persistent. Unfortunately, when trying to use the Persistent name ID format in OIF, it generated an error that “No service provider found.” This issue was clearly a bug due to the fact that there was a service provider configured and it worked for Transient.

In WebLogic under the Security Realms section->Providers->Authorization->ASIAuthorizationProvider->Attributes, after clicking the “Attributes” tab, the screen would crash. This issue occurred sporadically. Sometimes the attributes could be entered and viewed and other times the screen would display an error. After multiple WebLogic restarts, the error would eventually go away. It gives the user of WebLogic the sense that the product is unstable. The good news is once you enter the attributes and don’t make any changes to them, the attributes will consistently work as expected.

OIF provides a Service Provider Integration Module that displays a name of “Test SP Engine.” This provides a login screen that can authenticate a user through LDAP. This login screen provides exactly what we need for a production environment because we don’t need to integrate the authentication with an Identity Management System. However, Oracle named this engine “Test SP Engine.” This makes our development team question if this is suitable to be used in a production environment because it has the name test in it. Our guess as to why it is called “Test” is that Oracle wants us to use their Oracle Single Sign-on or Oracle Access Manager because they are provided without the word “test” in it. They do provide a custom integration engine tab but it makes no sense to make a custom LDAP authentication engine when the “Test SP Engine” works. Oracle should have called it “LDAP SP Engine.”

When attempting to create an Oracle HTTP server, we received the error “The specified module could not be loaded.” After a futile attempt to find an answer to the problem on Metalink, our team found the solution on a blog. We needed to copy a windows file called “msvcp71.dll” to the windows/system32 folder. Some versions of Windows don’t have the file included. A couple of months after our team found the solution on a blog, it was listed in Metalink.