SafeNet Authentication Service
Oracle Access Manager Manager
Configuration Guide
Document Information

<table>
<thead>
<tr>
<th>Document Part Number</th>
<th>007-012555-001, Rev. A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Date</td>
<td>September 2014</td>
</tr>
</tbody>
</table>

Trademarks

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording, or otherwise, without the prior written permission of SafeNet, Inc.

Disclaimer

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet invites constructive comments on the contents of this document. These comments, together with your personal and/or company details, should be sent to the address or email below.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>4</td>
</tr>
<tr>
<td>Purpose of this Guide</td>
<td>4</td>
</tr>
<tr>
<td>Applicability</td>
<td>4</td>
</tr>
<tr>
<td>Audience</td>
<td>4</td>
</tr>
<tr>
<td>Support Contacts</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 1 Overview</td>
<td>6</td>
</tr>
<tr>
<td>Authentication Flow</td>
<td>7</td>
</tr>
<tr>
<td>Environment</td>
<td>7</td>
</tr>
<tr>
<td>OAM Authentication Plugin Design</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 2 Plugin Installation and Configuration</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 3 Integrating the SafeNet Plugin Solution with Oracle Access Manager</td>
<td>12</td>
</tr>
<tr>
<td>Forcing the Authentication Scheme on a Protected Resource</td>
<td>20</td>
</tr>
<tr>
<td>Configuring the WebLogic Server</td>
<td>21</td>
</tr>
<tr>
<td>CHAPTER 4 Advanced Settings</td>
<td>24</td>
</tr>
<tr>
<td>Deploying Applications on Different Servers</td>
<td>24</td>
</tr>
<tr>
<td>Important INI Setting</td>
<td>25</td>
</tr>
<tr>
<td>Loading the INI File</td>
<td>25</td>
</tr>
<tr>
<td>Setting File and Directory Permissions</td>
<td>26</td>
</tr>
<tr>
<td>WebLogic Server SSL Errors (SSL connection to SAS)</td>
<td>26</td>
</tr>
<tr>
<td>Login Flow</td>
<td>28</td>
</tr>
<tr>
<td>Manually Removing the Plugin</td>
<td>29</td>
</tr>
</tbody>
</table>
Preface

Purpose of this Guide

This document describes how to configure SafeNet Authentication Service (SAS) Oracle Access Manager Agent. It contains the following chapters:

- “Overview” - Page 6
- “Plugin Installation and Configuration” – Page 10
- "Integrating the SafeNet Plugin Solution with Oracle Access Manager” – Page 12
- “Advanced Settings” – Page 24

Applicability

The information in this document applies to:

- **SafeNet Authentication Service (SAS)** — A cloud authentication service of SafeNet, Inc.
- **SafeNet Authentication Service – Service Provider Edition (SAS-SPE)** — The software used to build a SafeNet authentication service.
- **SafeNet Authentication Service – Private Cloud Edition (SAS-PCE)** — A term used to describe the implementation of SAS-SPE/PCE.

Audience

This document is intended for personnel responsible for maintaining your organization’s security infrastructure. This includes SAS users and security officers, the key manager administrators, and network administrators. It is assumed that the users of this document are proficient with security concepts.

All products manufactured and distributed by SafeNet, Inc. are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.
Support Contacts

If you encounter a problem while installing, registering, or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or SafeNet Customer Support. SafeNet Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

<table>
<thead>
<tr>
<th>Contact Method</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
<td>SafeNet, Inc.</td>
</tr>
<tr>
<td></td>
<td>4690 Millennium Drive</td>
</tr>
<tr>
<td></td>
<td>Belcamp, Maryland 21017</td>
</tr>
<tr>
<td></td>
<td>USA</td>
</tr>
<tr>
<td><strong>Phone</strong></td>
<td>United States</td>
</tr>
<tr>
<td></td>
<td>1-800-545-6608</td>
</tr>
<tr>
<td></td>
<td>International</td>
</tr>
<tr>
<td></td>
<td>1-410-931-7520</td>
</tr>
<tr>
<td><strong>Technical Support</strong></td>
<td><a href="https://serviceportal.safenet-inc.com">https://serviceportal.safenet-inc.com</a></td>
</tr>
<tr>
<td><strong>Customer Portal</strong></td>
<td>Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the SafeNet Knowledge Base.</td>
</tr>
</tbody>
</table>
CHAPTER 1
Overview

Oracle Access Manager provides APIs that allow software developers to write custom programs or components that integrate closely with Oracle Access Manager. These modules may represent anything from custom extensions of base Oracle Access Manager functionality to significant applications that are outside of Oracle Access Manager, but need to interact with Oracle Access Manager for identity or access control functions.

Authentication is the process of proving that a user is who he or she claims to be. The access system enables you to configure authentication rules in the policy domains and policies that protect your resources. Authentication rules, in turn, contain authentication schemes, which provide the methods for performing verification of a user’s identity.

This software utilizes OAM External Context Authentication Schema for two-factor authentication (2FA).

When a user tries to access a protected resource, Oracle Web Gate intercepts the request. It then redirects the request based on configured schema to an appropriate authentication destination. In this case, it’s a custom J2EE application (for example, CRYPTOCardOamLogin) deployed on the same web server (for example, WLS).

The application receives the request and collects the user’s credentials. It then sends the collected credentials to the cloud using highly encrypted (1024-bit) payload over SOAP protocol. The cloud then sends back an encrypted authenticated response. When the 2F authentication is successful, the user is redirected back to Web Gate along with the user’s LDAP credentials. Web Gate validates the LDAP credentials and on success redirects the user to the required destination. In case of failure, the user is redirected back to 2F authentication. If the user is already 2F authenticated, he can retry entering LDAP credentials (the number of unsuccessful attempts is configured in OAM). In OAM, this method of authentication is called chained authentication.

The Oracle Access Manager solution consists of the following applications:

- Access Manager (OAM)
- Web Logic Server (WLS)

The SafeNet OAM solution consists of the following applications:

- Authentication plugin (protects against unauthorized access attempts even with correct LDAP credentials)
- Collector and 2F authentication application

The overall authentication flow works with chaining the following OAM built-in plugins:

- User Identification plugin
- User Authentication plugin
This system of authentication in OAM is called “chained authentication,” which is depicted below:

### Authentication Flow

- The user accesses the protected application.
- OAM redirects the user to Collector and the two-factor authentication application (2FA).
- The SafeNet web application authenticates the user and redirects them to OAM with an encrypted security token.
- The SafeNet authentication plugin receives the request, authenticates and validates the encrypted security token, and then creates the credentials required by a built-in “user identification plugin.”
- The built-in user identification plugin identifies the user against a configured identity store and, on success, passes the information to the built-in “user authentication and authorization plugin”.
- OAM grants access and redirects the user to the required resource.

### Environment

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Platforms</td>
<td>- RedHat 5.7 64-bit</td>
</tr>
<tr>
<td></td>
<td>- Oracle Linux x64</td>
</tr>
<tr>
<td>Additional Software Components</td>
<td>Java Runtime (JRE)</td>
</tr>
</tbody>
</table>
OAM Authentication Plugin Design

The OAM Authentication Plugin design is as follows:

- **User ID** field with OTP authenticated session ID
- LDAP password with OTP authenticated session ID
- User IP addresses
- “request_id” parameter sent by OAM (in DCC mode only)
- Timestamp indicating when redirect happened
- Ticks (in milliseconds) indicating when redirect happened
- OTP authenticated session ID
- Success URL (Optional)

**NOTE:** Due to technical limitations in the loading and execution framework plugin, Java-based AES encryption is not available. Therefore, for encryption, the agent uses the existing encryption algorithm and key used to encrypt SAS communication by the agent.

When the Authentication plugin receives the security tokens, it decrypts the tokens, separates all the fields, and validates them as follows (any failure in decryption will trigger a failure message):

1. The **User** and **Password** fields are decrypted and the session ID is extracted. The session ID will be compared in the blob. A failure in the decryption and a non-matching session ID triggers a rejection.
2. The agent expects all parameters in the decrypted blob (if any one of them is missing, authentication will be rejected).
3. The OAM “request_id” and the OAM “request_id” session are compared. Non-matching ID’s will trigger an authentication rejection. This field is not available in ECC mode; therefore, the system checks to see if it exists.
4. The timestamp/ticks in the decrypted blob are used to calculate the time span or interval that occurred between the OTP authenticated redirect and when the agent received the parameters.
5. The maximum time for the acceptance of the token is controlled via the **REDIRECT_TIME_TO_LIVE_IN SECONDS** parameter in the agent INI file. The default setting is 2 seconds.
6. The session ID taken from the decrypted blob will be compared with the actual session ID. Any difference between the two will cause a rejection.
7. The client IP address will be checked, and if it is missing, authentication will be rejected.
8. The referrer header will be checked, it must not be missing. A missing referrer header means that it is the first request (sent by MITM) and did not originate from the OTP authenticated client.
9. If everything defined above is accepted, the credentials are passed to the built-in OAM User Identification plugin in the correct format (for example, in the format it must be received to validate them).
Follow the steps below to install and configure the plugin.

**NOTE:** Before proceeding with the configuration, make sure you back up your OAM configuration. In case you have any issues, you can always roll back OAM/WLS configuration files to restore OAM to its original state. All XML files can be located at:

/opt/oracle/Middleware/user_projects/domains/OAMdomain/config/fmwconfig

The `oam-config.xml` file must be backed up. You can find both applications on the same physical server.

To install the plugin:

1. Run the following command:
   ```bash
   rpm -ivh cryptocard-oam-agent-[your installation build].x86_64
   ```
2. The installation package is installed in `/usr/local/cryptocard/oam`. 
3. Configure the **INI** file that is present in the `\ini` sub-folder:

   ![INI file configuration screenshot]

4. Double-click the **JCryptoWrapper.ini** file.

5. In the editing tool that opens, configure the following keys:
   - **PrimaryProtocol**=https
   - **PrimaryServer**=Primary SAS Server Host OR IP
   - **PrimaryServerPort**=443

   If you have a secondary failover system, the settings below are optional:
   - **SecondaryProtocol**=https
   - **SecondaryServer**=Secondary SAS Server Host OR IP
   - **SecondaryServerPort**=443

   **NOTE:** If your organization uses a proxy server to access extranet or intranet, you must also configure the proxy settings in the INI file. The agent software works only with HTTP proxy (basic or anonymous authentication). Any settings changed (at any time) require a WLS and OAM restart.
This chapter provides information on how to integrate SafeNet's plugin solution to Oracle Access Manager (OAM).

To integrate the plugin:

1. Launch the OAM management web portal.
2. Enter your administrator username and password, and then click Login.

![Oracle Access Manager Login](image)
3. Click the **System Configuration** tab, and then click **Common Configuration**.

4. Double-click **Plugins**.

5. On the **Plugin** tab, click **Import Plugins**.

6. On the **Import Plugins** window, click **Browse** to locate the relevant plugin file in the **war** folder.

7. On the **File Upload** window, browse to the folder `/usr/local/cryptocard/oam/war`.

8. Select the file **CRYPTOCARDOamAuthModule.jar** and then click **Open**.
9. On the **ImportPlugin** window, click **Import**. The plugin appears as **Uploaded** in the **Plugins** list.

![Image of the Import Plugin window]

10. Select the uploaded **CRYPTOCardOamAuthModule** file from the **Plugins** table.

11. Click **Distribute Selected** to distribute the plugin to all OAM servers. If the plugin status does not change to **Distributed**, wait a few seconds and then click **Refresh** (you may have to refresh several times).

12. Click **Activate Selected** to activate the plugin. If the plugin status does not change to **Activated**, wait a few seconds and then click **Refresh** (you may have to refresh several times). Once the status appears as **Activated**, the plugin is ready to be used with the default settings.
NOTE: If the plugin is not deployed in the default location, change the required plugin parameters as shown previously and set the appropriate parameters in the INI file.

13. Click the System Configurations tab, and then click the Access Manager Settings menu. Expand Authentication Modules.

14. Click and expand Custom Authentication module, and then click the Create icon. A new Custom Authentication Module tab opens.
a. In the Name field, enter SafeNet
b. In the Description field, enter SafeNet Authentication Custom Module.

15. Click the Steps tab, and then click the icon to add a new first step in the authentication flow. The Add new step window opens.

Complete the following fields, and then click OK:

<table>
<thead>
<tr>
<th>Step Name</th>
<th>OTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Description</td>
<td>OTP Validation Step</td>
</tr>
<tr>
<td>Plug in Name</td>
<td>Select CRYPTOCardOamAuthModule.</td>
</tr>
</tbody>
</table>
16. Click the icon again to add a second step in the authentication flow.

![Authentication Module](image1)

Complete the following fields, and then click **OK**:

<table>
<thead>
<tr>
<th>Step Name</th>
<th>LDAP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Description</td>
<td>User Identification Step</td>
</tr>
<tr>
<td>Plug in Name</td>
<td>Select UserIdentificationPlugin.</td>
</tr>
</tbody>
</table>

17. Click the icon again to add a third step in the authentication flow.

![Authentication Module](image2)

Complete the following fields, and then click **OK**:

<table>
<thead>
<tr>
<th>Step Name</th>
<th>LDAP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step Description</td>
<td>User Authentication Step</td>
</tr>
<tr>
<td>Plug in Name</td>
<td>Select UserAuthenticationPlugin.</td>
</tr>
</tbody>
</table>
The **Steps** tab displays the three steps added above.

<table>
<thead>
<tr>
<th>Step Name</th>
<th>Description</th>
<th>Plugin Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTP</td>
<td>OTP Validation Step</td>
<td>CTPDatoCiscoSafeNetModule</td>
</tr>
<tr>
<td>LDAP1</td>
<td>User Identification Step</td>
<td>UserIdentificationPlugin</td>
</tr>
<tr>
<td>LDAP2</td>
<td>User Authentication Step</td>
<td>UserAuthenticationPlugin</td>
</tr>
</tbody>
</table>

18. Select the **Steps Orchestration** tab and select all fields in the table as indicated in the image below, and then click **Apply**.

A new **Custom Authentication Module** appears in the list on the left.
19. Click the **Policy Configuration** tab.

20. Select **Shared Components** and then **Authentication Schemes**.

21. Click the **Create** icon in the command bar.
Complete the following fields, and then click **OK**:

<table>
<thead>
<tr>
<th>Name</th>
<th>CRYPTOCardAuthenticationScheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>CRYPTOCard Authentication Scheme</td>
</tr>
<tr>
<td>Authentication Level</td>
<td>2</td>
</tr>
<tr>
<td>Default</td>
<td>This option should not be selected.</td>
</tr>
<tr>
<td>Challenge Method</td>
<td>FORM</td>
</tr>
<tr>
<td>Challenge Redirect URL</td>
<td>/oam/server</td>
</tr>
<tr>
<td>Authentication Module</td>
<td>SafeNet</td>
</tr>
<tr>
<td>Challenge URL</td>
<td>Enter SafeNet credentials and 2F authentication For example, in SafeNet’s DCC based system, this URL was: <a href="http://iamdemo.oracle.com:14100/Login/Login.jsp">http://iamdemo.oracle.com:14100/Login/Login.jsp</a> For ECC application, you can use only “/Login”.</td>
</tr>
<tr>
<td>Context Type</td>
<td>External</td>
</tr>
<tr>
<td>Challenge Parameters</td>
<td>Leave blank.</td>
</tr>
</tbody>
</table>

22. Click **Apply** to save the Authentication Scheme.

**Forcing the Authentication Scheme on a Protected Resource**

**To force the authentication scheme on a protected resource:**

1. Select the **Policy Configuration** tab.
2. In the left pane, select **Application Domains > Apache22 > Authentication Policies > Protected Resource Policy**.
3. Click the **Authentication Scheme** drop-down arrow and select **CRYPTOCardOamAuthenticationScheme** to force the policy. This scheme can be forced on any number of protected resources under OAM.
4. Click **Apply**.

---

**Configuring the WebLogic Server**

1. Log in to your Linux server hosting WebLogic Server (WLS) using SSH.
2. Add the **LD_LIBRARY_PATH** to the WLS startup scripts in the following locations:
   
   **Script 1:**
   
   `<WLS INSTALL PATH>/oracle/Middleware/wlserver_xx/common/bin/commEnv.sh`
   
   **Script 2:**
   
   `<WLS INSTALL PATH>/oracle/Middleware/user_projects/domains/OAMdomain/bin/setDomainEnv.sh`

   Remember to back up these files before modifying them.
3. Modify **script 1**, and at the end just before **resetFd**, add the following and save the file:

   ```
   LD_LIBRARY_PATH=/usr/local/cryptocard/oam/bin/x64:${LD_LIBRARY_PATH}
   export LD_LIBRARY_PATH
   ```

4. Modify **script 2**, and at the end just after **export BEA_JAVA_HOME**, add the following and save the file:

   ```
   LD_LIBRARY_PATH="/usr/local/cryptocard/oam/bin/x64":${LD_LIBRARY_PATH}
   export LD_LIBRARY_PATH
   ```

5. Shut down the Oracle Access Manager and WebLogic Server, and then restart the WebLogic Server.
6. Log in to the WLS web interface and, in the left pane, select **Deployments**.

![Oracle WebLogic Server Administration Console](image)

7. Click **Install**.

8. Select **CRYPTOCARDOMLogin.war**, and then click **Next**.

9. Select **Install this deployment as an application**, and then click **Next**.

10. Select the **oam_server1** check box, and then click **Next**. Click **Next** again in the **Install Application Assistant** window.

11. Click **Finish**. To complete the deployment process.

12. Start the WLS server. The deployed application will be active in the WLS.

If it is not active, go to the WLS and OAM log files to see what caused the error. Failure messages are sent to **std out**.
13. The SafeNet log file will be generated by default in the following location:

    /usr/local/cryptocard/oam/log/

**NOTE:** If the **LD_LIBRARY_PATH** environment variable is not defined, the WLS or OAM fails.

14. Add the IP address of the OAM Agent computer to the SAS Auth Node table as follows:
   a. Select the **Comms** tab, and then click **Auth Nodes**.
   b. Do not set this Auth Node as FreeRADIUS sync-enabled.
15. Click **Apply**.
Deploying Applications on Different Servers

Deploy the installation package on both servers and then proceed with configuring them. Make sure you configure the **Challenge URL** correctly in the **Authentication Schemes** tab. It must be an absolute URL to the SafeNet external Collector and two-factor authentication web application.

---

**NOTE:** Some INI file settings control the Server SSL Certificate Validation and logging level. You must go through these settings and adjust them accordingly. The login page images and captions are also in the INI file along with error messages. They must also be changed accordingly.
Important INI Setting

The following must be adjusted as per your network requirements for additional security reasons:

; Time to live value in seconds. Its time period between redirect and data passed to
; OMA LDAP authentication process. Can be reduced to few seconds may be 4-5 based on network latency
; and level of security required.
REDIRECT_TIME_TO_LIVE_IN_SECONDS=2
; If set to 1, SOAP calls to SAS will be made using Java functions
; If set to 0, SOAP calls will go through C++ Layer using JNI (Slow as compared to Java)
USE_JAVA_FOR_SOAP_CALLS=1

By default it’s 2 seconds. It’s a time period between web application redirects 2FA authenticated user to OAM and it reach Custom SafeNet Authentication Plugin. If both 2FA web application and OAM are on same WLS, it should be 1-2 seconds. It could be increased otherwise.

NOTE: If the applications are on different physical servers, the server’s clocks must be accurate and synchronized in order for authentication to be successful.

Loading the INI File

You can load the INI file from a location other than the default.

If the installation was installed in a location other than the default, you can define –DINI_PATH=<Full Path to the INI file including name> in OAM and WLS launch scripts. You can add this to the JAVA_OPTIONS environment variable in setDomainEnv.sh.

For Example:
JAVA_OPTIONS="-DUseSunHttpHandler=true -DINI_PATH=/usr/local/myloc/JCryptoWrapper.ini"
CHAPTER 4: Advanced Settings

NOTE: If you intend on using your own location, then remove the INI file from the default location.

Setting File and Directory Permissions

Administrators must secure file and directory access permissions. For example, READONLY access to the INI file must be restricted and limited to the WLS or OAM process owner. Plugin and web applications will require READONLY permissions for the INI file, key file, and resource directory, as well as files in that directory. The log directory must have WRITE permissions for the WLS/OAM process owner.

WebLogic Server SSL Errors (SSL connection to SAS)

As the certificate policy is different in WebLogic than it is in a stand-alone Java program, it is advised to use a Sun implementation instead of WebLogic.

http://webtech-kapil.blogspot.com/2010/06/javalangclasscastexception.html

Start WebLogic with the following flag:

-DUseSunHttpHandler=true

The standard Sun SSL implementation will be used.
When SOAP transport in the INI file is via Java code, on WLS you might encounter SSL connection, certificate validation, and SSL handshake errors. These errors can be prevented by adding the following to your WLS startup script below the comments block:

```
<WLS INSTALL PATH>/oracle/Middleware/user_projects/domains/OAMdomain/bin/setDomainEnv.sh

# added to avoid SSL errors
JAVA_OPTIONS="-DUseSunHttpHandler=true"
```
Login Flow

You can access protected resources once you have completed all of the necessary configurations. OAM will redirect you to the SafeNet web application:

Try logging in using all the following use cases:

**No User Name:**

**No LDAP Password:**

**Incorrect OTP:**

**Incorrect LDAP Password:**
All OK:

Test submitting correct LDAP credentials (direct access via good LDAP credentials) while another tab holds the valid OAM redirect. This penetration test case was successful in previous plugin versions but now is blocked by the custom SafeNet authentication plugin.

Manually Removing the Plugin

If you have backed up your XML files, a system rollback may be performed by restoring the configuration files. Optionally, you can manually remove the plugin by deleting the plugin node from the `oam-config.xml` file.

To manually remove the plugin:

1. Log in to the OAM console as an administrator.
2. Under **Policy Configuration > CRYPTOCardAuthenticationScheme**, change the **Authentication Module** field from **SafeNet** to **LDAP**.

3. Click the red x on the command bar to delete the **SafeNet custom authentication module**, as highlighted in the image below.
4. Click **Deactivate Selected**, and then click **Remove Selected**.

5. Shutdown the OAM and Admin Server (WLS).

6. Delete the file `CRYPTOCardOamAuthModule.jar` from the following locations (some locations only exist in specific versions of OAM R1):
   - `/opt/oracle/Middleware/user_projects/domains/OAMdomain/config/fmwconfig/oam/plugins`
   - `/opt/oracle/Middleware/user_projects/domains/OAMdomain/oam/plugin_repository/*`
   - `/opt/oracle/Middleware/user_projects/domains/OAMdomain/oam/plugins`

7. All sub dirs. Under `felix-cache oam` or `plugincache` (bundle*):
   - `/opt/oracle/Middleware/user_projects/domains/OAMdomain/felix-cache_oam`
   OR
   - `/u01/oracle/Middleware2/user_projects/domains/IDMDomain/servers/oam_server1/oam/plugincache`

8. Edit the `oam-config.xml` file and search for the text `CRYPTOCardOamAuthModule`. If it exists, delete the following:

```xml
<Setting Name="CRYPTOCardOamAuthModule" Type="htf:map">
  <Setting Name="name" Type="xsd:string">CRYPTOCardOamAuthModule</Setting>
  <Setting Name="author" Type="xsd:string">uid=kmushtaq</Setting>
  <Setting Name="email" Type="xsd:string">support@safenet-inc.com</Setting>
  <Setting Name="creationDate" Type="xsd:string">10:00:00, 2014-04-05</Setting>
  <Setting Name="version" Type="xsd:integer">2</Setting>
  <Setting Name="type" Type="xsd:string">Authentication</Setting>
  <Setting Name="source" Type="xsd:string">Custom</Setting>
  <Setting Name="jarFileName" Type="xsd:string">CRYPTOCardOamAuthModule.jar</Setting>
  <Setting Name="description" Type="xsd:string">CRYPTOG OAM Custom Authentication Plugin</Setting>
  <Setting Name="interface" Type="xsd:string">oracle.security.am.plugin.authn.AuthenticationPlugIn</Setting>
  <Setting Name="implementation" Type="xsd:string">CRYPTOCardOamAuthModule</Setting>
  <Setting Name="checksum" Type="htf:map">UiuwD775qH6Pj7eNJ8lQYENbQ=</Setting>
  <Setting Name="initiator" Type="xsd:string">Administrators</Setting>
  <Setting Name="stateFlags" Type="htf:map">
    <Setting Name="upload" Type="htf:map">
      <Setting Name="initiateDate" Type="xsd:string">03:29:59 2014-05-20</Setting>
    </Setting>
  </Setting>
</Setting>
```
9. Restart the WLS and then the OAM. Note that if you incorrectly delete the above node, the OAM will not start.