IBM Security Policy Manager 7.1: Using Access Manager Credentials

White Paper

July 2012

Ori Pomerantz
(orip@us.ibm.com)

Security Intelligence.
Think Integrated.

Smarter security solutions from IBM

Table of contents

Introduction

In this white paper, you learn how to transfer credential information from Access Manager for eBusiness to IBM WebSphere Application Server using TAI++. From IBM WebSphere Application Server, you then transfer the credential information to IBM Security Policy Manager.

Audience

This white paper is designed for security administrators who need to transfer credential information.

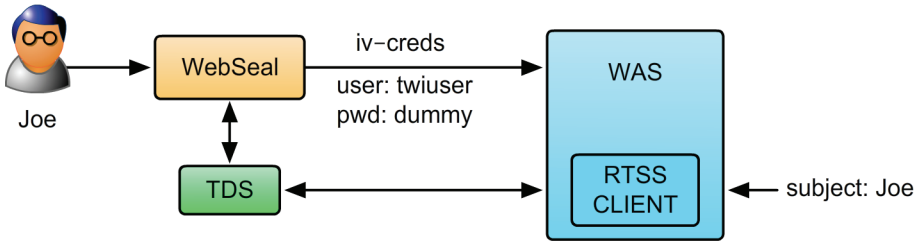
Acknowledgments

I would like to thank Norman Field for his help in setting up TAI++ and Craig Forster for fact checking the white paper. Any remaining errors are, of course, my responsibility.

1 WebSphere Application Server configuration

To send the credential information from Tivoli Access Manager for eBusiness to WebSphere Application Server, use ***TAI++ authentication***. Then, have WebSphere Application Server decode the credential and send it to Security Policy Manager.

When accessing a back-end server with TAI++ authentication, Tivoli Access Manager for eBusiness sends a constant user name and password. The server authenticates using those fields, and then decodes the **iv-creds** HTTP header to obtain the actual credential information for the user. This information is propagated to the RTSS client for authorization decisions in the **subject** category of the XACML.



This white paper shows the basic steps to setup support. More information is available at the following web address:

```
http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/
index.jsp?topic=%2Fcom.ibm.tspm.doc_7.1%2Fconfig%2Ftask%2Fco
nfiguringWASforTAI.html
```

1.1 Installing the TAI++ plug-in

WebSphere Application Server 7.0 comes with two TAI plug-ins (Tivoli Access Manager and SPNEGO). However, even the Tivoli Access Manager interceptor is limited to the user identity. To see all the credential fields, you need to install and configure the TAI++ plug-in.

___ 1. Download the plug-in from `http://www-01.ibm.com/support/entdocview.wss?uid=swg24016601` and extract it.

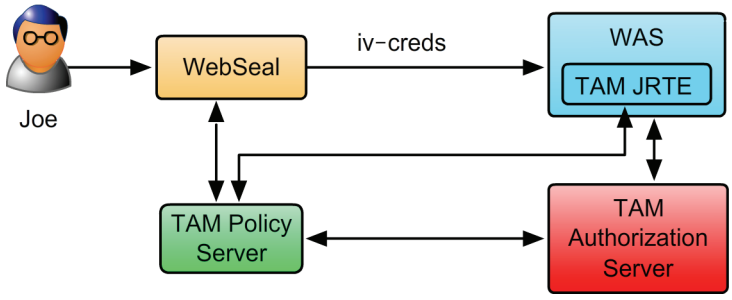
___ 2. Source the WebSphere Application Server environment for the profile that runs the RTSS client.

___ 3. Copy the plug-in to the WebSphere Application Server plug-in directory.

```
cp com.ibm.sec.authn.tai.etai_6.0.jar $WAS_HOME/plugins
```

1.2 Configuring the Tivoli Access Manager Java run time on WebSphere Application Server

The function that decodes **iv-creds** is part of the Tivoli Access Manager runtime environment. To use it from WebSphere Application Server, install and configure the Tivoli Access Manager Java runtime environment (JRTE).



___ 1. Register the run time with the Tivoli Access Manager policy server (all one line):

```
$WAS_HOME/java/jre/bin/java \
   -Dpd.cfg.home=$WAS_HOME/java/jre \
   com.tivoli.pd.jcfg.SvrSslCfg -action config \
   -admin_id sec_master -admin_pwd <password> \
   -appsvr_id was \
   -policysvr <TAMeb server host name>:7135:1 \
   -authzsvr <TAMeb server host name>:7136:1 \
   -port 999 -mode remote \
   -cfg_file $WAS_HOME/java/jre/PdPerm.properties \
   -key_file $WAS_HOME/java/jre//PdPerm.ks \
   -cfg_action create
```

For additional information about this step, see the IBM Education Assistant at this web address:
http://publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp?topic=/com.ibm.iea.tam/tam/6.1/using_api/connect_to_policy_server/connect_to_policy_server_viewlet_swf.html.

___ 2. Get the Tivoli Access Manager server list.

```
pdadmin -a sec_master -p <your password> server list
```

```
tspm:~/tai++ # pdadmin -a sec_master -p object00 server
  ivacld-tspm.tivoli.edu
  was-tspm.tivoli.edu
  default-webseald-tspm.tivoli.edu
```

1.3 Configuring WebSphere Application Server to accept TAI++ authentication

___ 1. Browse to the Integrated Solutions Console (ISC).

___ 2. Log on as an administrator.

___ 3. Click **Security > Global security**.

___ 4. Click **Web and SIP security > Trust association** on the right-side of the window.

___ 5. Click **Interceptors**.

___ 6. Select the two default interceptors and click **Delete**.

___ 7. Click **New**.

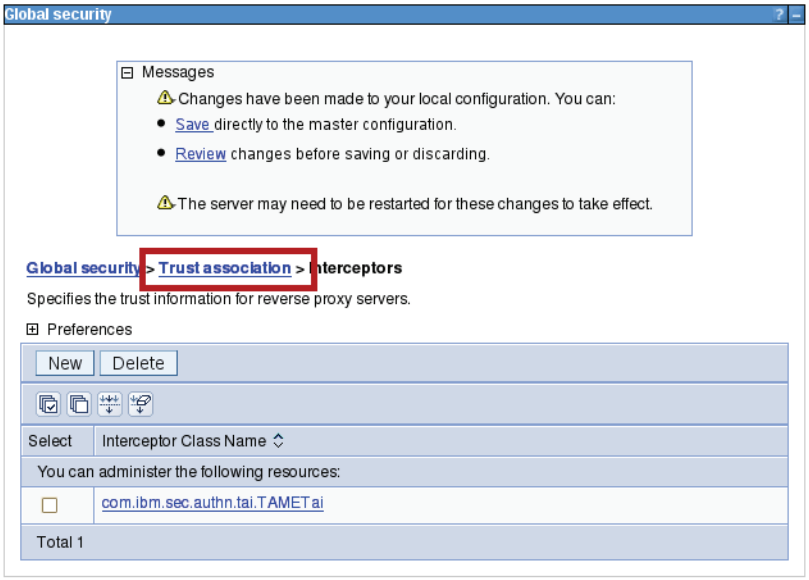
___ 8. Specify the class name **com.ibm.sec.authn.tai.TAMETai**.

___ 9. Specify these custom properties. Click **New** for additional lines.

Name	Value
com.ibm.websphere.security.webseal.configURL	<your WAS home>/java/jre/PdPerm.properties
com.ibm.websphere.security.webseal.id	iv-creds
com.ibm.websphere.security.webseal.loginId	taiuser

___ 10. Click **OK**.

___ 11. Click **Trust association**.



___ 12. Select **Enable trust association** and click **OK**.

___ 13. Click the **Save** link near the top of the browser window.

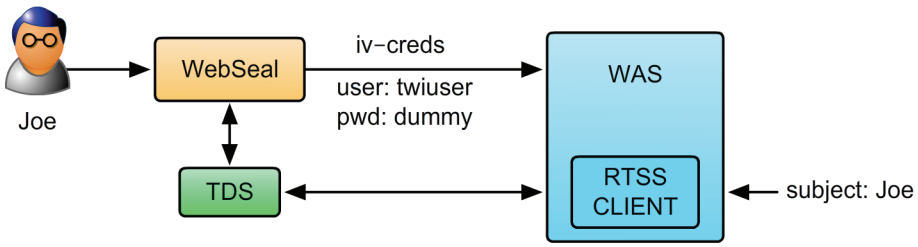
___ 14. Restart the WebSphere Application Server.

2 Tivoli Access Manager for eBusiness configuration

Next, configure Tivoli Access Manager for eBusiness to authenticate as a TAI user, and send the credential in to WebSphere Application Server in the HTTP header.

2.1 Creating a TAI++ user

Users log on to WebSEAL using their own credentials. WebSEAL then sends a dummy user name and password to log on to WebSphere Application Server. It also sends the credential in the **iv-creds** HTTP header field, which can then be decoded by WebSphere Application Server.



___ 1. Identify the dummy password used by the WebSEAL instance. If you use the default WebSEAL instance, this is the command:

```
cd /opt/pdweb/etc
grep basicauth-dummy-passwd webseald-default.conf
```

___ 2. Create the user:

```
pdadmin -a sec_master -p <sec_master passworD>
u c taiuser uid=taiuser,<prefix> tai usr <dummy password>
u m taiuser account-valid yes
quit
```

2.2 Configuring a TAI++ junction

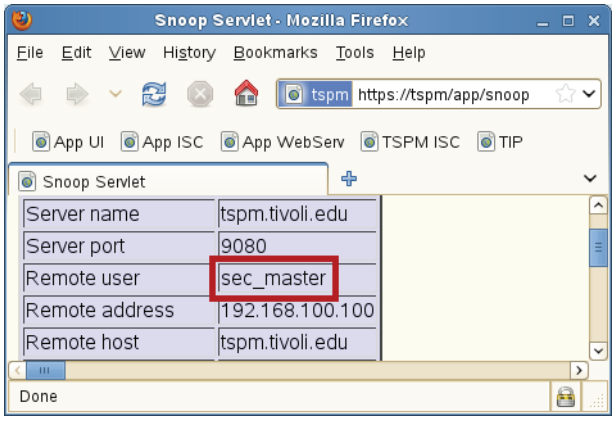
___ 1. Create the junction (all one line):

```
pdadmin -a sec_master -p <password> \
   s t <WebSEAL instance> create -t tcp \
   -h <WAS host> -p <port> -b supply -c iv-creds /tai
```

___ 2. Restart the browser and log on through Tivoli Access Manager to WebSphere Application Server:

```
https://<WebSEAL>/tai/snoop
```

___ 3. Log on as an authorized user. See that the **Remote user** field contains the Tivoli Access Manager user.

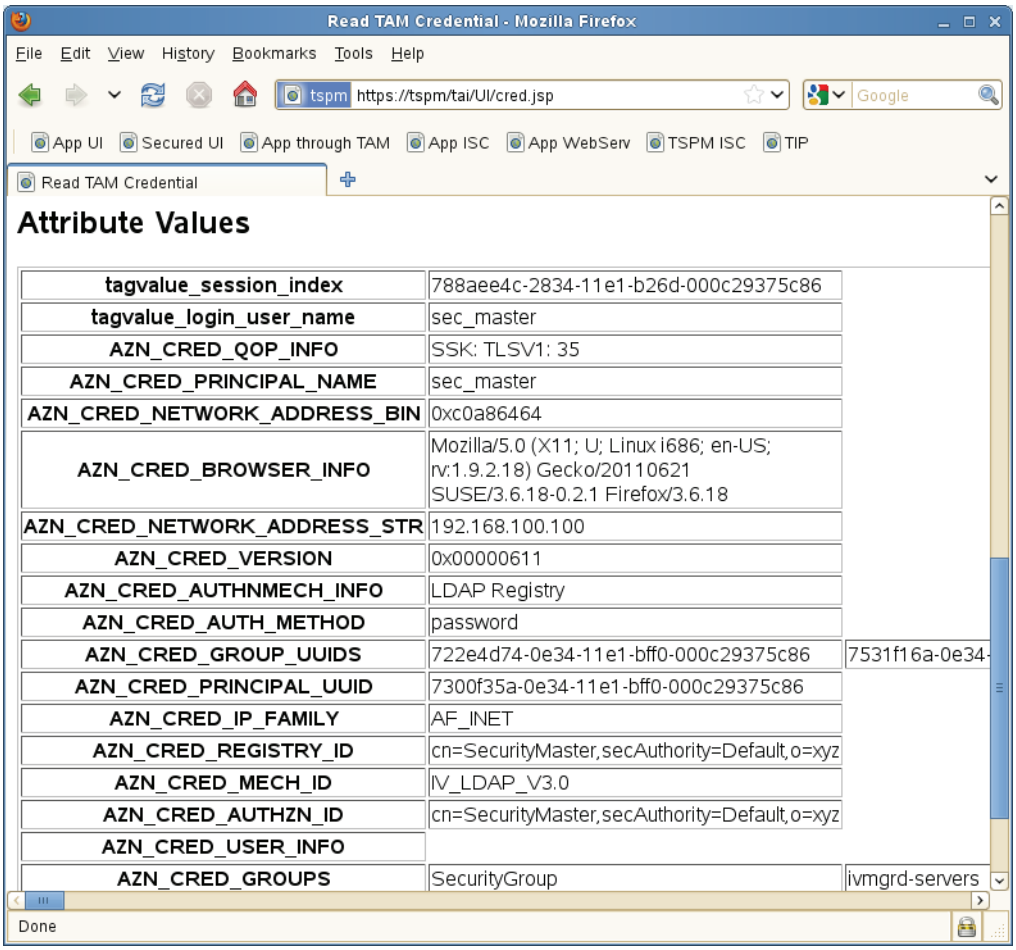


3 Using Tivoli Access Manager for eBusiness credential fields in JSP

Use the Tivoli Access Manager for eBusiness API to decode the value in **iv-creds**. This step is explained in great detail in the following web address:

```
http://www.ibm.com/developerworks/tivoli/tutorials/tz-tamauthapi/index.html
```

The **cred.jsp** code in appendix A produced this screen capture. You can read it to see how to use the credential fields in your own code.



4 Using Tivoli Access Manager for eBusiness credential fields in Security Policy Manager

After you verify the credential fields are transferred correctly to WebSphere Application Server, use them in Security Policy Manager to make authorization decisions.

4.1 Configuring WebSphere Application Server to interpret the credential

In this step you configure WebSphere Application Server to use the Tivoli Access Manager for eBusiness API to interpret the **iv-creds** header, similar to the way **cred.jsp** does. For more information about this topic, see the following web address:

```
http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/
topic/com.ibm.tspm.doc_7.1/config/reference/
updatingconfigxml.html
```

___ 1. Change to the correct directory and create a backup copy.

```
cd $CONFIG_ROOT/cells/$WAS_CELL/commonauthz
cp config.xml config.xml.old
```

___ 2. Open **config.xml** in an editor. Change the highlighted text from false to **true** to enable Tivoli Access Manager integration. Then, restart WebSphere Application Server.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CommonAuthzConfig>

   <Config owner="XMLStore" enabled="true" name="XMLStoreFactory" context="com.ibm.sec.authz.int
ernal.storage">
       <Attributes>
           <Attr name="providerClass"  value="com.ibm.tscc.rtss.argus.RTSSXMLStore" />
           <Attr name="providerPlugin" value="com.ibm.tscc.rtss.argus.RTSSXMLStore" />
       </Attributes>
   </Config>

   <Config owner="TAM session integration" enabled="true" name="tam" context="com.ibm.sec.authz.
jaccplus">
       <Attributes>
           <Attr name="configURL" value="file:///opt/IBM/WebSphere/AppServer/java/jre/PdPerm.pro
perties" description="The TAM configuration file"/>
           <Attr name="sendToken" value="false" description="Extract BinarySecurityToken with iv
creds"/>
           <Attr name="sendAttributes" value="true" description="Extract TAM session attributes"
/>
        </Attributes>
   </Config>

</CommonAuthzConfig>
```

4.2 Using credential parameters in Tivoli Security Policy Manager policies

Create a string parameter and use it in a policy normally. When you configure the policy, select **Rule parameter is contained in the request, in a standard location** and select the **subject** section. The field names for Tivoli Access Manager for eBusiness credentials are documented at the following web address:

http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itame.doc%2Fam611_web_devref37.htm&path%3D3_4_6_4_3

Appendix A: The cred.jsp file

```
<HTML>
<HEAD>
<TITLE>Read TAM Credential</TITLE>
<%@ page language="java"
        import = "com.tivoli.pd.jazn.PDAuthorizationContext"
        import = "com.tivoli.pd.jazn.PDPrincipal"
        import = "java.lang.String"
        import = "java.net.URL"
        import = "com.tivoli.pd.jutil.PDAttrs"
        import = "java.util.Iterator"
        import = "com.tivoli.pd.jutil.PDAttrValue"
%>
</HEAD>

<BODY>
<% String cred = request.getHeader("iv-creds"); %>

<H2>Perliminary Data</H2>

<UL>
<LI> Raw credential: <%= cred %> </LI>
<LI> Locale: <%= request.getLocale() %> </LI>
<% String tamCfg =
      "file:///<directory>/PdPerm.properties"; %>
<LI> TAM cfg: <%= tamCfg %> </LI>
<% PDAuthorizationContext tamContext = new
PDAuthorizationContext(
        request.getLocale(), new URL(tamCfg));
%>
<LI> PDAuthorizationContext: <%= tamContext %> </LI>
<% PDPrincipal tamUser = new PDPrincipal(tamContext,
cred.getBytes()); %>
<LI> PDPrincipal: <%= tamUser %> </LI>
<% PDAttrs attrs = tamUser.getAttrlist(tamContext); %>
</UL>

<H2>All Attributes</H2>

<OL>
<%
for (Iterator i = attrs.keySet().iterator(); i.hasNext() ;)
{
String attr = (String) i.next();
%>
<LI><%= attr %></LI>
<% } %>
</OL>
```

```
<H2>Attribute Values</H2>
<TABLE BORDER>
<%
for (Iterator i = attrs.keySet().iterator(); i.hasNext() ;)
{
String attr = (String) i.next();
%>
<TR><TH><%= attr %></TH>
<%
for (Iterator ii = attrs.getValues(attr).iterator();
ii.hasNext() ;) {
%>
<TD>
<%= (String) ((PDAttrValue) ii.next()).getValue() %>
</TD>
<%
}
%>
</TR>
<% } %>
</TABLE>

</BODY></HTML>
```

