Guide to Enabling Single Sign-on for Flows Started with the Java Flow Invoke Tool

Issued on September 12, 2007

This Guide is intended for customers, Opsware Systems Engineers (SEs), and Customer Engineers (CEs) who have installed or are deploying Opsware PAS 7.0.

Enabling single sign-on for flows started with the Java Flow Invoke tool

You can obtain security and performance benefits by configuring Central so that flows that are started from the Java version of the flow invocation tool (JRSFlowInvoke.jar) use the credentials of the person who is already logged on the machine. This is called *single sign-on (SSO).*

Note: SSO support in Central is based on the standard Kerberos 5. The procedures for enabling single sign-on for Central vary depending on whether Central is to use a Linux key distribution center (KDC) or a Windows KDC (Active Directory, which supports the Kerberos 5 specification). These procedures are documented in the following two sections, which assume that the reader is familiar with Kerberos fundamentals, that is, terms such as principal, ticket, realm, KDC and keytab.

Enabling single sign-on using Windows AD

To track an example through the following procedure, we'll assume the following:

- Central (either Windows or Linux) is located at alamo.mydomain.com
- The Windows AD domain controller is at mydomain.com
- The realm is MYDOMAIN.COM (note that for Windows AD, the realm name is usually the domain name, upper-cased).
- The account for which SSO is attempted is "jdoe".
- The PAS home directory is represented as "PAS_HOME" in discussion and in commands.

To enable single sign-on using Windows AD

1. Add an AD account for the host (the Central server that the Java flow invocation tool will point at when running the flows). The account must have the following format:

   `HTTP/<server_name.domain_name>`

   It is advisable to configure this AD account with the settings "Password never expires" and "Use DES encryption types for this account".

   If you do not set DES encryption types for the account, AD uses the RC4-HMAC encryption type.

Using our example, the account that you add would be:

HTTP/alamo.mydomain.com

2. On the domain controller machine, open a command-line window and generate a keytab file, using the following command:

```
ktpass -out <server_name>.keytab -princ
<service_name>/<server_name.domain_name>@<REALM_NAME> -mapuser
<service_name>/<server_name.domain_name> -pass *** -crypto DES-CBC-
MD5 –ptype KRB5_NT_PRINCIPAL
```

where:

*** is the password that you specified when you created the above AD account.

In our example, this command would look like this:

```
ktpass -out alamo.keytab -princ HTTP/alamo.mydomain.com@MYDOMAIN.COM
-mapuser HTTP/alamo.mydomain.com -pass *** -crypto DES-CBC-MD5 –
ptype KRB5_NT_PRINCIPAL
```

Copy the keytab file (alamo.keytab in our example) to the Central server, into PAS_HOME/Central/conf directory.

3. Open PAS_HOME/Central/conf/jaasLogin.conf in a text editor.
4. Add the following “com.sun.security.jgss.accept” section after the DharmaKrb5JAAS section, replacing PAS_HOME in the highlighted section with the correct path:

```
DharmaKrb5JAAS {
   com.sun.security.auth.module.Krb5LoginModule required
               refreshKrb5Config=true;
};

com.sun.security.jgss.accept {
        com.sun.security.auth.module.Krb5LoginModule
         required
         storeKey=true
         doNotPrompt=true
         useKeyTab=true
         kdc=mydomain.com
         keyTab="PAS_HOME/Central/conf/alamo.keytab"
         realm="MYDOMAIN.COM"
         principal="HTTP/alamo.mydomain.com@MYDOMAIN.COM"
         debug=true;
};
```

5. In Central/conf, create a krb5.conf file that includes definition of the default realm and KDC (or make sure that the existing krb5.conf includes that information).

In our example, a minimal krb5.conf file would look like this:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000
```

```
[realms]
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
        default_domain = .mydomain.com
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

6. Log in to Central and, on the **Administration** tab, click the **System Configuration** subtab.
7. Scroll down to **Kerberos Authentication Settings** and configure the location for the Kerberos 5 configuration file (krb5.conf) to point to "/Central/conf/krb5.conf".

   Notes:

   - Do not set a realm or a KDC on that page, because Central will now obtain them from the krb5.conf file.
   - You do not need to enable Kerberos authentication unless that is used for logging in.
8. Save your changes, and then restart Central.

   By default, under PAS_HOME/Central/tools (where the java flow invocation tool JRSFlowInvoke.jar is installed) there is an sso_invoke_krb5.conf.sample file that looks like the following:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000


[realms]
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
        default_domain = .mydomain.com
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

9. Copy sso_invoke_krb5.conf.sample to sso_invoke_krb5.conf and edit the latter to match your domain, realm, and KDC.

   By default, under PAS_HOME/Central/tools there is an sso_invoke.bat file for the Windows Central version (or sso_invoke.sh for the Linux Central version) that shows how to use the java flow invocation tool in single sign-on mode. You can run those shell scripts from that location. Or, if the invocation tool is to be used from a different machine than the Central server, copy the JRSFlowInvoke.jar, sso_invoke.bat (or sso_invoke.sh), and sso_invoke_krb5.conf files to that machine and adjust the paths (including the path to JRE 1.6, which is required on the target machine—you can obtain JRE 1.6 from the downloads page of the Java site, http://java.sun.com/).

   You can invoke the shell scripts with a command such as the following:

   ```
   sso_invoke alamo.mydomain.com:8443 /Library/MyFlows/myFlow
   ```

10. Log in to Central with an account that has Administrator rights.

    Next, you will need to give HEADLESS_FLOWS capability to the SSO users.

11. The easiest way to give HEADLESS_FLOWS capability to the SSO users is:
    a. In Central, on the **Administration** tab, click the **System Configuration** sub-tab.
    b. Scroll to the Kerberos section and set the default group to a group that has HEADLESS_FLOWS capability.

       This way, any headless invocation using SSO will have the capabilities of that group (flows cannot be invoked using the headless tool unless the user under whose credentials the invocation happens, has HEADLESS_FLOWS capability).

    Or, if SSO flow invocations need to be controlled on a user-by-user basis:

    - On the **Administration** tab, create the Central user that matches the account under which the SSO flow invocation will happen (“jdoe” in our example) and specify that it is an external user.

      For information on how to create a user and specify that it is an external user, see Help for Central.

      The user must be a member of a group that has HEADLESS_FLOWS capability; without this capability, the user will not be able to start runs using SSO flow invocation.

    In addition to having the HEADLESS_FLOWS capability, the user under whose credentials the SSO flow invocation happens needs to have **read** and **execute** permissions for the flow and the operations that the flow uses. For more information on granting permissions to flows and operations see Help for Studio.

12. If the SSO java invocation is from a Linux machine that is not configured to obtain Kerberos tickets automatically, obtain a forward-able ticket from the Windows domain controller (you might have to change /etc/krb5.conf to point it to the Windows domain controller), using a command like the following:

    ```
    kinit –f jdoe@MYDOMAIN.COM
    ```

13. If Central is a Windows version hosted on a Windows 2000/2003 system, add the following registry key (do the same for the machine where the java invocation tool is to invoked from, if the machine is Windows 2000/2003):

    ```
    HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
    Value Name: allowtgtsessionkey
    ```

```
Value Type: REG_DWORD
Value: 0x01
```

Enabling single sign-on using MIT KDC

Note: The following procedure assumes that the system uses a Linux version of MIT KDC.

To track an example through the following procedure, we'll assume the following:

- Central (either Windows or Linux) is located at fitzroy.mydomain.com
- KDC is at kdc.mydomain.com
- The realm is MYDOMAIN.COM.
- The account for which SSO is attempted is "jdoe".
- The PAS home directory is represented as "PAS_HOME" in discussion and in commands.

To enable single sign-on using MIT KDC

1. On the KDC machine, add a service principal for HTTP/fitzroy.mydomain.com@MYDOMAIN.COM using the kadmin's `addprinc` command (for information on using kadmin, see the man pages for kadmin):
   ```
   kadmin: addprinc –randkey HTTP/fitzroy.mydomain.com@MYDOMAIN.COM
   ```
2. Export the principal you just created to fitzroy.keytab:
   ```
   kadmin: ktadd –k fitzroy.keytab HTTP/fitzroy.mydomain.com
   ```
3. Copy the keytab file to the Central machine at PAS_HOME/Central/conf
4. In Central/conf, create a krb5.conf file that includes definition of the default realm and KDC (or make sure that the existing krb5.conf includes that information).

   In our example, a minimal krb5.conf file would look like this:

   ```
   [libdefaults]
           default_realm = MYDOMAIN.COM
           ticket_lifetime = 24000
           default_tkt_enctypes =  des3-cbc-sha1

   [realms]
       MYDOMAIN.COM = {
           kdc = kdc.mydomain.com
           admin_server = kdc.mydomain.com
           default_domain = mydomain.com
       }


   [domain_realm]
       .mydomain.com = MYDOMAIN.COM
       mydomain.com = MYDOMAIN.COM
   ```

```
[pam]
    debug = true
```

5. Open /Central/conf/jaasLogin.conf in a text editor.
6. Add the following “com.sun.security.jgss.accept” section after the DharmaKrb5JAAS section, replacing PAS_HOME with the correct path:

```
DharmaKrb5JAAS {
   com.sun.security.auth.module.Krb5LoginModule required
               refreshKrb5Config=true;
};

com.sun.security.jgss.accept {
        com.sun.security.auth.module.Krb5LoginModule
         required
         storeKey=true
         doNotPrompt=true
         useKeyTab=true
         kdc=kdc.mydomain.com
         keyTab="PAS_HOME/Central/conf/fitzroy.keytab"
         realm="MYDOMAIN.COM"
         principal="HTTP/fitzroy.mydomain.com@MYDOMAIN.COM"
         debug=true;
};
```

7. Log in to Central and on the **Administration** tab, click the **System Configuration** subtab.
8. Scroll down to **Kerberos Authentication Settings** and configure the location for the Kerberos 5 configuration file (krb5.conf) to point to "/Central/conf/krb5.conf".

   Notes:

   - Do not set a realm or a KDC on that page, because Central will now obtain them from the krb5.conf file.
   - You do not need to enable Kerberos authentication unless that is used for logging in.
9. Save your changes, and then restart Central.

   By default, under PAS_HOME/tools (where the java flow invocation tool JRSFlowInvoke.jar, is by default installed) there is an sso_invoke_krb5.conf.sample file that looks like:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000
        default_tkt_enctypes =  des3-cbc-sha1

[realms]
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
```

```
        default_domain = .mydomain.com
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

10. Copy sso_invoke_krb5.conf.sample to sso_invoke_krb5.conf and edit the latter to match your domain, realm and KDC.

    By default, under PAS_HOME/tools there is an sso_invoke.bat file for the Windows Central version (or sso_invoke.sh for the Linux Central version) that shows how to use the java flow invocation tool in single sign-on mode. You can run those shell scripts from that location. Or, if the invocation tool is to be used from a different machine than the Central server, copy the JRSFlowInvoke.jar, sso_invoke.bat (or sso_invoke.sh), and sso_invoke_krb5.conf files to that machine and adjust the paths (including the path to JRE 1.6, which is required on the target machine—you can obtain JRE 1.6 from the downloads page of the Java site, http://java.sun.com/).

    The shell scripts can be invoked with a command such as in the following:

    ```
    sso_invoke fitzroy.mydomain.com:8443 /Library/MyFlows/myFlow
    ```

11. Log in to Central with an account that has Administrator rights.

    Next, you will need to give HEADLESS_FLOWS capability to the SSO users.

12. The easiest way to give HEADLESS_FLOWS capability to the SSO users is:

    c. In Central, on the **Administration** tab, click the **System Configuration** sub-tab.

    d. Scroll to the Kerberos section and set the default group to a group that has HEADLESS_FLOWS capability.

       This way, any headless invocation using SSO will have the capabilities of that group (flows cannot be invoked using the headless tool unless the user under whose credentials the invocation happens, has HEADLESS_FLOWS capability).

    Or, if SSO flow invocations need to be controlled on a user-by-user basis:

    - On the **Administration** tab, create the Central user that matches the account under which the SSO flow invocation will happen (“jdoe” in our example) and specify that it is an external user.

      For information on how to create a user and specify that it is an external user, see Help for Central.

      The user must be a member of a group that has HEADLESS_FLOWS capability; without this capability, the user will not be able to start runs using SSO flow invocation.

    a. In addition to having the HEADLESS_FLOWS capability, the user under whose credentials the SSO flow invocation happens needs to have **read** and **execute** permissions for the flow and the operations that the flow uses. For more information on granting permissions to flows and operations see Help for Studio.

13. If the SSO flow invocation is from a Linux machine that is not configured to obtain Kerberos tickets automatically, obtain a forward-able ticket from the KDC (you might have to change /etc/krb5.conf to point it to the kdc.mydomain.com in our example), using a command like the following:

    ```
    kinit –f jdoe@MYDOMAIN.COM
    ```

14. If the SSO flow invocation is from a Windows machine, a forward-able ticket needs to be obtained from the Linux MIT KDC. This can be done by using kinit executable under PAS_HOME/jre1.6/bin.
15. If the SSO flow invocation is from a Windows 2000/2003 system, add the following registry :

    ```
    HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
    Value Name: allowtgtsessionkey
    Value Type: REG_DWORD
    Value: 0x01
    ```

Enabling SSO when a network load balancer is used

The procedure is the same as in the above sections, the only change being that the service principal and keytab files are generated for the network load-balancer (NLB) machine and not for the individual Central nodes behind the load balancer.

For example, suppose that:

- The NLB machine is nlb.mydomain.com
- There are two Central nodes behind the load balancer: central1.mydomain.com and central2.mydomain.com

In this case, the service principal would be HTTP/nlb.mydomain.com@MYDOMAIN.COM (if Windows AD is used, the AD user account would be HTTP/nlb.mydomain.com), and the keytab file would be nlb.keytab.

In addition, you must:

- Copy the keytab to central1.mydomain.com and central2.mydomain.com.
- Modify the respective entries in jaasLogin.conf on those machines to point to keytab=nlb.keytab and principal=HTTP/nlb.mydomain.com@MYDOMAIN.COM

When you call the SSO flow invocation script, make sure that it points to nlb.mydomain.com, as in the following:

```
sso_invoke nlb.mydomain.com:<port_number> /Library/MyFlows/myFlow
```

where `<port_number>` is the port on which the network load balancer is listening.