René Mulder

[rene.mulder@forgerock.com](mailto:rene.mulder@forgerock.com)

19 November 2012

V1.0

# SSO with Citrix XenApp hosted by Solaris

## Introduction

### Contents

This document contains ForgeRock’s recommendation for implementing Single Sign-On for applications accessed through Citrix hosted on Solaris specific to the existing Ericsson environment.

It also contains a comparison between the recommended solution and the leading alternative.

Finally, there is a list of previously considered, but closed alternatives for completeness.

### Citrix Environment

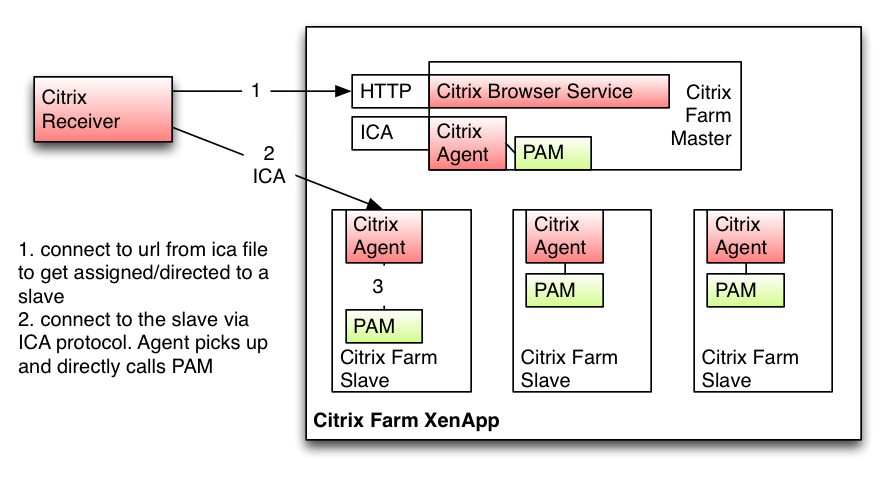
Ericsson has an OpenAM implementation and use OpenAM Policy Agents to secure access to web-applications.

They face the challenge of securely accessing Solaris Desktop applications through Citrix with Single Sign-On.

Citrix provides a proprietary mechanism for SSO, but that is not available since a key component of that Citrix SSO solution is not available. It is also key to realize that the Citrix web portal is not used. There is no existing SSO based on Citrix technology over the different applications. In the current environment, static, standard ICA filesare generated on a custom-built server. The ICA files point to the Citrix Farm server to assign the client to one of the Solaris based application servers (UAS).

The Citrix receiver directly connects to the Solaris machine and starts the application over the ICA protocol. The Citrix Agent uses PAM to authenticate the user, be it through a login screen or directly if the password is provided in the ICA file (insecure). The username is also used by an LDAP client to lookup user attributes such as their home directory. Modification of the applications running on Solaris is not an option due to their quantity and heterogeneity.

Since the Citrix mechanism is a closed system, there is no way to modify the red items in the diagram below. Even though it is possible to add additional parameters to the ICA file (such as tokens) there is no way to tell the Citrix Agent how to use these additional parameters.



### Existing SSO with OpenAM

The hyperlinks for accessing a myriad of applications are available after logging in to the Ericsson portal (named TOR Launcher).The TOR launcher is protected by OpenAM through a web-agent placed on the Apache HTTP reverse proxy server. The hyperlinks point to both web-applications and ICA files. The web-applications have OpenAM agents on them to provide SSO.At the moment the ICA files are static, meaning their content doesn’t change per session. They are generated by the ICA builder that has an OpenAM agent as well. Reverse proxies are abstracted from the diagrams, but are present.

## Problem statement

The challenge is to re-use the OpenAM session set up by logging into the portal when accessing / starting up an application on Solaris through Citrix.

### Line of reason

TheICA file is now static. It can be shared and reused and is therefore insecure. For an SSO scenario we need to introduce some sort of token (to prevent session hijacking) and do back-end verification. The first place available to do this is in the PAM configuration.

The attributes in the ICA file that the Citrix Agent uses for authentications is fixed and limited[[1]](#footnote-2). Of the limited set, we can only use the username and password as they are passed on to PAM.

We must to use the password attribute to store the token. We can’t use the username for this, since the username is required by to look up session information from LDAP.

## Solutions

### Recommended Solution

The ICA Builder injects the OpenAM token (iPlanetDirectoryPro) into the password field of the ICA file.

On the Solaris machine, we will introduce a new PAM module that validates the token. Validation is done by calling the OpenAM token validation REST interface:

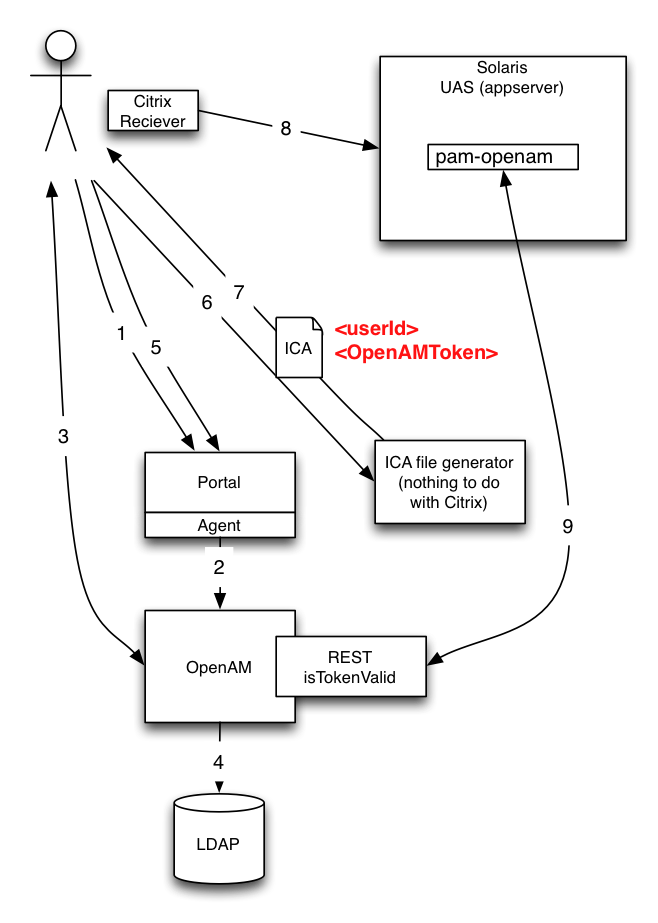
<https://openam.example.com:8080/openam/identity/isTokenValid?>

 tokenid=\*

The PAM module pam\_openam.so will be of type *auth* and be made *sufficient.* If no password is given, the module will report a failure to authenticate and not prompt for a password.

This setup will support two scenarios as required:

1. SSO login. The password field is supplied and contains the OpenAM token. The pam\_openam module will succeed and thus the PAM stack will succeed. pam\_ldap module will not be invoked.
2. Legacy login. The password field is supplied, but contains the LDAP password. pam\_openam will fail, but pam\_ldap will use the password and handle this unchanged
3. Legacy login. No password is supplied, pam\_openam will fail, but pam\_ldap will prompt for a password.



1. Browse to TOR launcher
2. Agent redirects to OpenAM
3. Enter credentials
4. Validate credentials
5. Redirect back to TOR launcher, access granted
6. Click on Citrix based application. ICA file is generated by ICA builder. iPlanetDirectoryPro token is available through HTTP header and injected in the ICA file
7. ICA file is downloaded to the client
8. Citrix receiver uses ICA file to establish connection
9. PAM module validates the token to OpenAM and grants access.

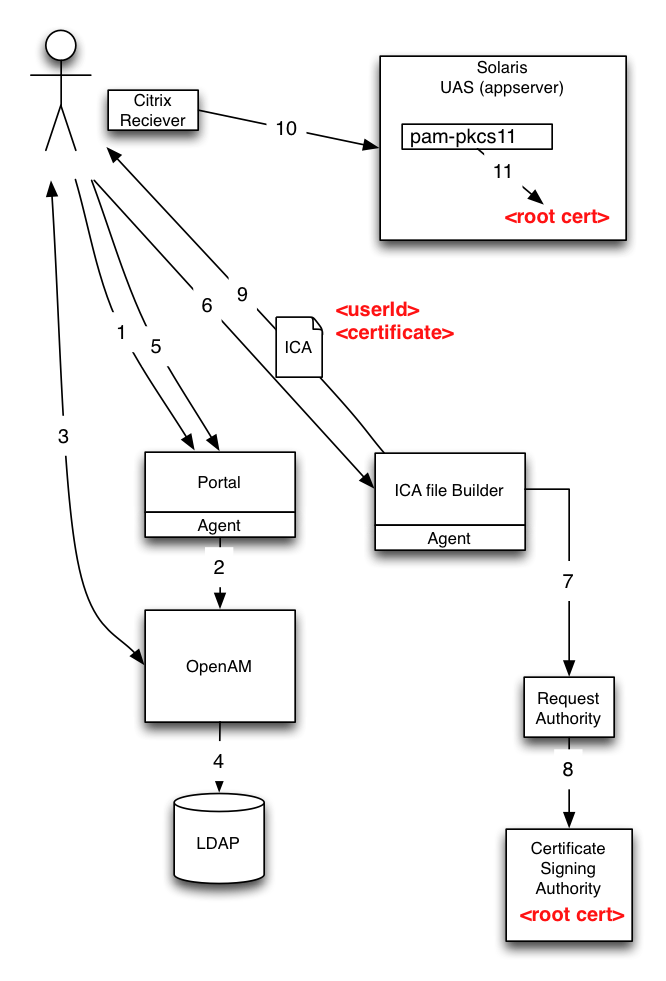
### Alternative solution

Generate a signed, short-lived X.509 certificate on the ICA builder and use that as the token. Use of short lived certificates is a GRID practice[[2]](#footnote-3). Verify the certificate with pam\_pkcs11 and assert that the subject equals the username.

pam\_pkcs11[[3]](#footnote-4) is an existing pam module that can verify X.509 certificates is. It can’t verify certificates to online CAs so the CA root certificate has to be placed on the machines beforehand. Online revocation lists are supported.

The ICA builder will create a certificate signing request for a certificate with a very short validity time (i.e. 1 minute) and offer that to the request authority, which in turn will deliver a signed certificate to be included in the ICA file. Alternatively, to reduce dependencies in this PoC scenario the RA and SCA is replaced by having the root certificate on the ICA builder and generating signed certificates itself.

Since access to the ICA builder is protected by OpenAM, this solution provides secure SSO.



1. Browse to TOR launcher
2. Agent redirects to OpenAM
3. Enter credentials
4. Validate credentials
5. Redirect back to TOR launcher, access granted
6. Click on Citrix based application. ICA file is generated by ICA builder
7. Create a CSR with subject <userId>
   1. Alternative: generate a certificate with subject <userId>
8. CA signs the certificate
   1. Alternative: ICA file builder signs certificate
9. ICA file is downloaded to the client
10. Citrix receiver uses ICA file to establish connection
11. PAM module validates the certificate and grants access.

### Closed alternatives

1. Introduce a Citrix STA server and use Citrix SSO
   1. The STA server does not expose an API, so the token can’t be requested by the ICA builder
2. Reverse engineer the Citrix STA server and implement it with an API and on the preferred OS
   1. Although feasible reverse engineering is against the Citrix license agreement
3. Replace Citrix with Secure Global Desktop
   1. Not feasible within required time frame
   2. Citrix has a unique selling point (hardware metrics based load balancing)
4. Introduce federation by making the ICA builder a Service Provider and use the pam module CrudeSAML[[4]](#footnote-5) to re-use the SAML response
   1. CrudeSAML can only trust one SP entityId. There are multiple SPs since there are multiple ICA builders
   2. Getting support for CrudeSAML is a challenge
   3. Hard to fetch the raw SAML response at the right time to be able to inject it in the ICA file
   4. Maturity of CrudeSAML is a concern
   5. Breaks SAML standard (misuse of SP target)

## Comparison

The comparison of the recommended and alternative solution is presented side by side since advantages for one are disadvantages for the other and vice versa.

|  |  |  |
| --- | --- | --- |
|  | **Recommended** | **Alternative** |
| Complexity | + The OpenAM token is short and simple  + Fewer components  -Requires a back-channel of the PAM module to OpenAM | - Based on the experience of members of the ForgeRock team, implementations based on X.509 certificates are often considerably more complex than initially assessed  - Introduction of an additional component (CA composed of RA and CSA) with all the associated system management and support requirements.  + Token validation is offline |
| Support | + ForgeRock can support the call-out from PAM to OpenAM and thereby include the Solaris Servers as part of the OpenAM based SSO solution  o Ericsson must support their bespoke PAM module | - ForgeRock can’t support the Solaris servers as they are technically a separate SSO domain.  o ForgeRock expects that since the uncommon manner of using the pam\_pkcs11 module, Oracle will not by default support it. |
| Scalability | + OpenAM sessions are discarded after expiry | - Management of certificates has been reported not to scale very well beyond 100,000 certificates per CA (depending on the CA implementation). This number will easily be reached over time since every ICA file generation will lead to a new certificate |
| Performance | + Performance of the REST API is proven to be very high (2.000 calls/s.). Sessions are lighter in creation and destruction. | o Generation and signing of certificates requires more computation. In a synchronous environment the requirement are high and considerable infrastructure might be required to accommodate large amounts of parallel generation. Performance tests should be employed to assure meeting requirements after the PoC. |
| Risk | + Although this exact solution has not been seen by ForgeRock before, all components of it (including backchannel use of REST) are in regular production use for other customers. | - There is no implementation experience within Ericsson/ForgeRock with this specific use of short lived certificates. This introduces unknowns, which are a risk. |
| Cost | + There is no additional cost in the support contract with ForgeRock regardless of the solution. The only change will be in the scope of the support (see support) | o ForgeRock expects that any perceived savings in avoiding custom development will be outweighed by the testing activities to ensure the certificate-based solution performs to requirements. |
| Standard | o The OpenAM token is not an open standard. ForgeRock does not see this as a disadvantage since the solution is already based on OpenAM and policy agents. In practice, when moving away from an existing solution, even when based on a standard, use of that standard is usually reconsidered and often revised anyway. | + X.509 is an open standard |
| Maintenance | - Additional maintenance on the code of the custom PAM module  - Maintenance of OpenAM URL on the PAM module | - Additional maintenance on the CA  - Management of root certificates and expiry of them |
| Development | - Development of a custom PAM interface | - Development of a request authority client (or a CA in the PoC situation) |

1. <http://support.citrix.com/proddocs/topic/ica-settings/ica-settings-wrapper.html> [↑](#footnote-ref-2)
2. <http://www.tagpma.org/authn_profiles/slcs> [↑](#footnote-ref-3)
3. <http://rpm.pbone.net/index.php3/stat/4/idpl/17636979/dir/rawhide/com/pam_pkcs11-0.6.2-8.fc17.x86_64.rpm.html> [↑](#footnote-ref-4)
4. <http://ftp.espci.fr/pub/crudesaml/README> [↑](#footnote-ref-5)