Test procedures for LSS

Table of contents

Version history

December 16th 2016	Version 2.0	TSS
4th April 2014	Version 1.1	JGB
28th March 2014	Version 1.0	JGB
20th March 2014	Version 0.92	JGB
13th March 2014	Version 0.91	BS
2014	Version 0.9	TSS
2014	Version 0.1	MSP

1 The purpose and audience of the document

This document includes test procedures that must be followed and reported by a LSS vendor for implementation and integration of the LSS for NemID API.

The document addresses the testers of a LSS for NemID implementation and integration setup.

Summary of all documents in the LSS for NemID package:

General documentation

- Terms and concepts in LSS for NemID

Implementation documentation

- General technical specification
- LSS technical specification
- Implementation guide for LSS

Test documentation

- Guidelines on the use of LSS for NemID test tools
- Recommended test procedures for LSS for NemID
- Testprocedures for LSS

Solution documentation

- Requirements feedback form for LSS

2 Introduction

For a Local Signature Server (LSS) supplier to be able to offer users LSS for NemID, the supplier must implement, set up and maintain a LSS for NemID back-end on the users' local network.

This document describes the test procedure for a LSS supplier integrating the LSS for NemID setup.

The LSS supplier's tests are divided into three parts. The first part is a functional test with the purpose of verifying whether integration has been completed successfully and works as intended. The second part is test focused, on a set of user-scenarios testing parts of integration. The second part requires that the first functional tests have been completed successfully.

The last test part includes testing of the LSS Client Network setup. These tests require that the DNS for the LSS endpoint[1] is working on the end-user's local network and that the trust for the (custom) SSL-certificate has been set up on end user devices.

[1] https://lss-for-nemid-server.dk and optional https://lss-for-nemid-server-test-dk

3 Test steps

The NETS service provider package automatically integrates to the LSS flow when using the CodeFile client for employee certificates (MOCES).

It is required that the LSS endpoint URLs are configured in local DNS or similar methods (ex. Windows host file) and that a trusted self-provisioned SSL certificate is configured and trusted by the client browsers.

The recommended test steps are outlined as follows:

1. Setup the test LSS back-end stub and configure local DNS and SSL to https://lss-for-nemid-server-test.dk.

2. Setup an example service provider page, using a service provider package for NemID.

3. Test that the LSS flows works. Note, that the "NemID Nøglefilsprogram" plugin available from NETS as a browser plugin needs to be disabled in the browsers used for LSS.

4. Switch the LSS back-end from the test stub to your LSS back-end implementation.

5. Complete the tests described in this document.

6. Test the production setup on the network. This includes DNS to the LSS Client URL and SSL-trust on end user devices.

4 Functional tests

This section describes the test procedure for validation of the technical parts of the NemID integration.

4.1 Test of HTML5 Web Messaging setup

The communication between the LSS back-end and the service provider, through the LSS Client is handled in JavaScript and by using the HTML5 Web Messaging API. The LSS supplier must test and verify that the JavaScript functions specified in the technical documentation has been setup and is working per the specification.

In addition, the LSS supplier tests that the back-end can parse the JSON packages received.

4.2 Handling the /<random digits> in the request

The service provider will request

https://lss-for-nemid-server.dk/<random-digists>

or

https://lss-for-nemid-server.dk/?t<random>

Test that the web server can handle this.

4.3 Identify erroneous JSON

Return the correct error code to the service provider, when the commands received contain improperly constructed JSON structures.

4.4 Missing parameter error code

Return the appropriate error code to the service provider, when a required parameter is missing.

As an example; a sign text parameter is required for signing a document.

4.5 Parameter case-insensitivity

The parameters received in the JSON structure are mapped in name-value pairs. The names of the values are case-insensitive. Test that this is handled correctly.

4.6 Validate time stamp

One of the parameters received from the service provider is a time stamp. The timestamp must not be more than 3 minutes old. Respond correctly on invalid values.

4.7 Validate signature on parameters

The service provider provides a signature on the parameters sent to the LSS back-end. The LSS back-end must validate the signature.

Test and verify that your setup can validate the signature. The signing certificate must be a valid VOCES or FOCES certificate issued by Nets DanID.

4.8 Validate digest over parameters

The LSS back-end must be able to calculate the exact same digest over the received parameters, as the one generated by the service provider.

4.9 Error codes

Test that the correct error codes are returned for all specific error codes.

To do so, end set up and test scenarios for all error codes specified in the LSS technical specification documentation.

The **LSSGLB001** error code scenario is not tested. This should never be returned by the LSS supplier.

4.10 Test of XML-DSig Message returned to the service provider

Verify that the XML-DSig Message returned to the service provider is valid per the technical specification. The service provider demo application provided in the LSS for NemID SP package can be used as a test service provider and by this to validate the XML-DSig Message. The service provider demo application uses OOAPI to validate the XML-DSig Message and certificates.

It is required that OOAPI validates the XML-DSig Message. The LSS supplier is free to choose either the Java or .Net variant of OOAPI.

5 Use case tests

Complete the same use cases as described in "SP Recommended test procedures". The same outcome is expected for all cases.

5.1 User with more than one certificate tries to log in

Preconditions: A setup with a LSS back-end configured with a user with more than one certificate of which one should be valid. The test stub provided in the TU package supports this.

1. The user enters the website of the service provider to access a self-service page.
2. The user finds the log-in button or link and activates it.
3. The user is presented to the NemID interaction design and selects the LSS for NemID option.
4. The user enters user name and password and chooses between his or her certificates. More than one should be available in the drop-down selector presented.
5. The user selects some certificate and proceeds to login.
6. The user repeats this process for at least two different certificates.

Result: The user is logged in or rejected as expected, based on the selected certificates.

6 Sign text validation and rendering

The requirements stated in DanID's documentation on the various sign-text flows are the minimal requirement when validating sign text as a LSS supplier.

For HTML, the LSS back-end must enforce the same whitelist of allowed html tags in the signature.

Xml is transformed using the accompanying transformation and then validated as html before shown to the user.

PDF is validated using the whitelist specified in DanID's documentation.

For text, no special validation is required. The LSS must check that the text displayed to the user are displayed as text and thus not run any dynamic content in any way.

6.1 Rendering

For text signing flow, remember to implement support with a mono-space font as this is a part of the API. Do not use word-wrap when displaying the sign-text when a mono-space font is used.

The PDF signing flow is not mandatory. If implemented - ensure that the document is rendered as expected on a range of supported devices. It is important that the user sees the PDF to-be-signed correctly and thus mechanisms to check and ensure that the user has the expected PDF displayed correctly must be enforced.

If the PDF fails to render correctly, the signing flow should not be allowed to complete successfully.

7 LSS Client DNS test

To use LSS for NemID, the user must be connected to a local network with local DNS for the LSS Client URL address https://lss-for-nemid-server.dk.

Test that DNS is setup and working such that clients connected to the local network can look up the LSS back-end using the LSS Client URL.

8 LSS Client SSL trust

The LSS Client is started using the LSS Client URL running over https. The LSS back-end must be hosted using custom SSL certificates for lss-for-nemid-server.dk.

Test and verify that the custom SSL certificates can be setup for the LSS Client URL and that trust for this certificate (i.e. for the CA for this certificate) can be installed on the user's devices and that content are shown and communication can be made with the LSS Client *iframe* when setup using the LSS Client URL.

9 Handling of ADDITIONAL_PARAMS

The parameter ADDITIONAL_PARAMS are an optional key/value list of parameters sent from the service provider to the LSS back-end. If the LSS back-end receives key/value pairs which are unknown to the LSS back-end they must be ignored.

If however, ADDITIONAL_PARAMS_CRITICAL are set, the LSS back-end must ensure that keys defined in ADDITIONAL_PARAMS_CRITICAL are

1. Check that all values received in PARAMS match a key in the received ADDITIONAL_PARAMS. If not, return the status code *LSSADP000*

2. If any value received in ADDITIONAL_PARAMS_CRITICAL is unknown or unrecognized by the LSS back-end, return the status code *LSSADP000*

Test and verify that the LSS back-end implementation returns the LSSADP000 status code as expected.