ESGF Security System Integration Test Plan

Security Meeting at Argonne National Laboratory, 14-16 September 2010

Attendees: Philip Kershaw, Rachana Ananthakrishnan, Gavin Bell, Eric Nienhouse, Luca Cinquini, Nathan Wilhelmi

Change Log

Version	Date	Author(s)	Change
0.1	13 Sept 2010	Philip Kershaw, Rachana Ananthakrishnan	Initial Draft
0.2	14-15 Sept 2010	Philip Kershaw, Rachana Ananthakrishnan, Gavin Bell, Eric Nienhouse, Luca Cinquini, Nathan Wilhelmi	ANL Meeting: fleshed out cross-cutting, interface based and walk-through tests
0.3	23 Sept 2010	Philip Kershaw	Revised structure, added deployment matrix
0.4	23 Sept 2010	Rachana Ananthakrishnan	Removed action items to separate spreadsheet, cleaned up notes

Contents

1. Objective

The objective of this plan is to verify a working federation. This is meant as distinct from unit and integration testing of individual software components developed for ESG. Rather it means the verification that all the deployed software at the different partner sites can correctly interoperate. Implicit then in this is a focus on the interfaces involved in site to site interactions. The ESGF Security ICD specifies what these are. All interfaces in all deployments should be tested and probed to ensure compliance with requirements outlined in the ICD and the requirements of the underlying specifications used. e.g. SAML, OpenID.

2. Methodology

2.1 Identify a Baseline

The starting point is to define what is the target system for testing:

- Which institutions are covered? – PCMDI, ORNL, NCAR, BADC, DKRZ, JPL.
- What components are deployed at each institution? The deployment matrix lists this information.
- What are the software versions and host environments?

2.2 Testing Approach

Adopt a layered approach to testing.

1. At the base level define tests based on specifications, protocols which cross-cut all components. These may be suitable for automation with test scripts.
2. Build on this with tests for the individual interfaces with emphasis on verification of services involving institution - institution interactions.
3. At the next level, walk-throughs – scenarios which probe the full functioning of the system. For example, a user registers for CMIP5 access, receives confirmation of membership and then access data from a Data Node secured with that authorisation role.

Failure at a given level of testing precludes continuation of any further higher level tests. There are multiple implementations of some interfaces - SAML and OpenID. Testing services deployed with the different implementations will provide a more robust test of the system. e.g. Python SAML client calls Java SAML service.

3. Test Cross Cutting Concerns

Cross cutting concerns where underlying technologies are in common, for example use of SSL. ESG mandates mutual authentication with SSL.

3.1 Trust root Repository Provisioning

Verify that each site has up to date:

- list of CAs and CRLs

- Whitelist of Gateways, Data Nodes, Attribute Services and Authorization Services

3.2 Basic System Test

- Verify that server host clocks are synchronized. This is important for timestamp checks in place for some interfaces.

3.3. X.509 Credentials

Pass criteria:

- Public key bit size – 2048
- SHA2 by next year (2011)

(MyProxy client side needs to generate these larger keys and use SHA2,
Security endpoint that use these certificates need to enforce this requirements,
Test the secure endpoint by querying it with a non compliant certificate)

3.4 SSL

Pass Criteria:

- Rejects connection from peer where no certificate is provided.
- Rejects peer certificates issued from untrusted CAs
- Rejects peer certificates not included in the appropriate white list
    - White list to be defined per service
- Rejects insecure renegotiation
- Accepts only SSLv3
- Rejects connections that are not privacy protected.

(This implies mutual authentication)

3.5 Generic SAML Tests

SAML in Authorization service and Attribute service.

Pass Criteria:

- Rejects any SAML over non-SSL connections
- Rejects any SAML request and response from entities not included in the appropriate white list
    - White list to be defined per service
- Reject SAML assertions without Subject NameID as urn:esg:openid
- Reject any SAML query that does not have a Request ID
- Reject any response that does not have a InResponseTo ID corresponding to the Request ID
- Reject SAML assertions that are not with in the NotBefore and NotOnOrAfter, and with configured check for clock skew tolerance of 15 seconds either way.

4. Individual Interface Tests

4.1 Provisioning

Verify that the Provisioning Service:

- Reject client requests from non-federation members - based on certificate DN
- Reject any clients that don't present a certificate from a trusted CA

Verify that Provisioning Clients:

- Add new entries for CAs and DN whitelists
- Remove local CAs and DNs that have been removed from the Provisioning Service
- Changes filter down to inidividual services which require SSL settings.

4.2 Authentication

4.2.1 OpenID Authentication

- OpenID Provider Tests:
    - Check that the OpenID is well formed URL
    - GET OpenID URL and check Yadis document content. Check Attribute Service and MyProxy URI endpoints are included along with the OpenID Provider one
    - Rejects non SSL request
    - Rejects incorrect login - incorrect username, incorrect password, null or empty password, non-existent username
    - Reject wrong username entered for the given OpenID
    - Reject non-SSL Relying Party
- Relying Party Tests
    - Rejects Identity Providers that don't use SSL
        - Test OpenID URL is SSL based, then parse the Yadis document and get the Provider endpoint and check that this endpoint is SSL.
    - Rejects OpenID Provider not on the whitelist
        - Specific whitelist to be provider

4.2.2 PKI Authentication

- MyProxy (PKI Credential Issuer)
    - Rejects incorrect login - incorrect username, incorrect password, null or empty password, non-existent username
    - Check that it returns a X.509 End Entity Certificate (EEC)
    - Issues credential only to users registered at the Gateway (How do we provision the MyProxy server endpoint on the web start?)
    - Only issues short term certificates - lifetime of the certificate should only be 12 hours
- PKI Relying Parties (Authentication Filters)
    - Rejects any credential from a untrusted CA
        - Specific list to be provided
    - Rejects any credential that has expired

4.2.3 Authentication Service

- Authentication assertions should issue assertions that are limited by the credential presented to the authentication service
- X.509 Certificate to OpenID specifics (ORP today)

- Cookie should be a session limited cookie
- Session lifetime should be limited to the credential lifetime

4.2.4 Authentication Service RPs

- Reject any authentication assertion that has expired
    - E.g. TDS that uses the ORP, should check that the cookie has expired.
- If the assertion does not have a expiration time, 12 hours is used as default.
    - E.g. If TDS is accessed by a user via a browser, the OpenID assertion should be set to 12 hours.

4.3 Attribute Exchange

4.3.1 Attribute Agreements

- Reject any extension with OID `1.2.3.4.4.3.2.1.7.8` in X.509 Certificate, that is not a SAML Attribute Assertion
- Reject any SAML Attribute Assertion that does not have the correct data type for controlled attributes
    - Attributes defined in ICD

4.3.2 Attribute Service

- Reject any clients not in a white list
- Reject any clients that don’t present a certificate from a trusted CA
    - White list defined

4.3.3 Attribute Service Clients

- Reject any Attribute Service that does not use SSL
- Reject any server not in a white list
- Reject any server that does not present a certificate from a trusted CA
    - White list defined in the trust root bootstrap
- Reject any invalid SAML Assertion (defined earlier)
- Reject any attribute not from a authorized attribute issuer
    - List to be provided

4.3.4 Attribute Propagation

- OpenID Attribute Exchange:
    - Test all Providers can return the first name, last name, e-mail address attributes
    - Test attributes have the correct types, as per ICD

4.4. Authorization

4.4.1 Authorization Service

- Reject any clients not in a white list
- Reject any clients that don’t present a certificate from a trusted CA
    - White list defined

4.4.2 Authorization Service Client

- Reject any Attribute Service that does not use SSL
- Reject any server not in a white list
- Reject any server that does not present a certificate from a trusted CA
    - White list defined in the trust root bootstrap
- Reject any invalid SAML Assertion (defined earlier)
    - check AuthzDecisionStatement decision values - Permit, Deny etc. - and decision type.

4.5 User Attribute Registration

This is covered in the walk-throughs.

4.6 Data Node Manager

- Test resolution to obtain the attribute service endpoint.
    - Configuration test for OpenID providers and attribute service provider test

4.7 Publisher Service (Gateway interface)

- Reject any requests to publish if the requester does not have the specified role. Refer to ICD (publisher)
- User can register at Gateway for publisher attribute for *that* Gateway
- (SSL goodness)
- Test access to the published data
    - user is not registered for that/those role(s) - they're denied access
    - user is registered - they're granted access

4.8 Publisher client

User can get a credentail from an IdP that is separate to the Gateway to which they are going to publish to.

- Rejects publisher servers that don't use SSL
- (All SSL goodness for validation)
- Test publisher client can publish dataset tagging it with given authorisation role(s). The policy in the authorisation service is updated and a client access request validates that that policy is in place i.e.
- Check publishing the same dataset with a different role - does it change the policy or is not permitted as an operation.

4.9 Protected Services

This refers to protected services using HTTP/S ESG Security Protocol (TDS, et. al.)

- Using wget without a certificate:
    - When requesting a open data resource, should get 200 HTTP status code
    - When requesting a protected data resource:
        - If wget is configured without redirects, should get a HTTP 30X
        - If wget is configured to follow redirects, should get a 401

- Using wget with a valid certificate:
    - When requesting a open data resource, should get 200 HTTP status code
    - When requesting a protected data resource:
        - if the user does not have the correct authorisation, should get a 403
        - if the user does have the correct authorisation, should get a 200
- If data server, reject any request to upload data

Note: In cases where applicable, this applies to both file downloads and OPeNDAP access

4.10 GridFTP Server

- Test that files outside dataset root is not accessible
- Test authentication - trusted CAs and establishment of client identity
- Test authorization - ensure that if client does not have access to a particular dataset, it is rejected.
- Reject any request to upload data

5. Walk-through Tests

We assume that OpenID registration has already taken place: this step is specific to a Gateway whereas these tests are intended to probe operations which test interfaces across institutions in the federation

5.1 User Registration for CMIP5 access role

Initial Conditions:

- User knows that they need to register with PCMDI to gain access to the CMIP5 data
- User is registered with an OpenID at an IdP in Federation. This IdP is not PCMDI

Steps:

1. User invokes the PCMDI CMIP5 Registration page, verify page is available
2. They sign in with OpenID and fill out the form, verify form submission
3. They receive e-mail notification that registration succeeded. Verify notification is received.

Pass criteria:

- User is notified of registration decision
- User must sign in with OpenID and is able to
- PCMDI OpenID RP correctly whitelists the user’s OP i.e. user can’t sign with a non-ESGF

5.2 User Is Denied Access for CMIP5 Data at the Gateway and follows link to PCMDI to Register

This is an alternative to the previous flow.

Initial Conditions:

- User doesn’t know that they are required to CMIP5 access
- User is registered with an OpenID at an IdP in Federation. This IdP is not PCMDI

Steps:

1. User discovery data at Gateway

2. User signs in with OpenID
3. Gateway denies access to file listing as user does not have CMIP5 attribute. Gateway interface displays a link to the PCMDI CMIP5 Registration page.

5.3 User Accesses CMIP5 data via the Gateway WGet Scripts and MyProxyLogon WebStart

This flow continues from either of the previous two.

Initial Conditions:

- User is signed with OpenID
- User is already registered for CMIP5 attribute

Steps:

1. Get credentials from MyProxyLogon WebStart
2. Download WGet script
3. Download data from DataNode using WGet script

Pass criteria:

- User can access MyProxyLogon WebStart link
- Interface provides some means for the user to know their MyProxy server location
- User can get a credential
- User can download WGet script
- MyProxyLogon functions with any MyProxy Server in the federation (in practice one or two samples). Trust roots downloaded for the MyProxy server are correct. MyProxy server listening on standard port. The criteria does not imply that a user can get credentials from any MyProxy server, just to verify that the MyProxy is up and working, and can interface with the client.
- WGet script and MyProxyLogon interoperate correctly i.e.Credentials and trust roots locations match with WGet script expected locations
- WGet script succeeds in download. DataNode TDS and security middleware functions correctly

5.4 User access data secured with CMIP5 access role direct from a DataNode

Initial Conditions:

1. User browses data catalogs on the TDS hierarchically, identifies files of interest
2. User is stopped by the access control system, and logs in.
3. User downloads files

Pass criteria:

- User can access with TDS

6. Deployment Matrix

Identify the software components deployed at the partner sites. This oriented toward the individual security interfaces rather than their parent packages i.e. Gateway or Data Node.

	PCMDI	NCAR	ORNL	JPL	DKRZ	BADC

Provisioning Service		N/A	N/A	N/A	N/A	N/A
Provisioning Service Client						
OpenID Provider						Python NDG Security 1.5.7
MyProxyCA						
Registration Service						
Attribute Service						
Authorization Service						
Authentication Filter (ORP and SSL based authentication)						
Authorisation Filter						
GridFTP						
Publisher Service						
Publisher Client						

7. Notes

1. Stress / Loading tests?
2. Basic system requirements and some way to document that.
3. QC Use cases from Bryan Lawrence
4. Standalone client tools to test these
5. Verify data node publication - tests (register, setup attributes)
6. Trust everyone in the federation.
    a. Can you trust additional entities?
7. Is there a single stack across all the federation? If there is, many of these tests can be made into unit tests, and probe only configurable pieces.
8. If we automate this as part of monitoring service, how would the credentials and test be setup.
9. Session cookies only on the server side, short term credentials and CRLs for revocations
    a. Do performance test to see how it affects user experience to redirect to the ORP each time the session cookie expires
10. Caching of attributes and authorization decisions
11. PCDMI Attribute Service is a critical service, and has high risk profile and should have scalability solutions and fail-over solution.

8. References

- ESGF Interface Control Document: http://esg-pcmdi.llnl.gov/esgf/esgf-security-interface-control-documents/
- Action Items from the meeting: https://spreadsheets.google.com/pub?key=0AoRQqJO52t1WdGZydWJMTHBKMkZjdk5hS3Z1V2tVRkE&output=html