GSI with OpenSSL

Vincenzo Ciaschini

EGEE-3 All-Hands

Prague, 4-7/11/08

Information Society and Media

Layout

- GSI/SSL Differences and Issues
- VOMS without Globus

GSI/SSL Differences and Issues

- **Issue 1: What is GSI?**
  - GSI = SSL(+ proxy) + (Delegation)
    - So, if you do not use Delegation, you are fully compatible with SSL
  - Well, almost but no.
  - GSI = SSL(+ proxy) + extra_message + (Delegation)
    - SSL means SSL3, not SSL2 nor TLS.
  - Where extra_message is:
    - '0' – No delegation.
    - 'D' – Delegation follows.
  - Your SSL server should expect this message if it wants to stay compatible with GSI
  - Your SSL client must send this message it if wants to speak to a GSI server.
  - But is not there the GSS_C_GLOBUS_SSL_COMPATIBLE flag?
    - No. Only works reliably with gss_*() calls, not with gss_assist_*() calls.
    - So, always send that message.

GSI/SSL Differences and Issues

- **Issue 2: What to do with proxies?**
  - Proxies are not normally considered valid certificates.
    - OpenSSL verification must be extended with a custom validator.
    - See the one in src/sslutils.
  - What kinds of proxies do you want to support?
    - GT2, GT3, or GT4?
    - Globus only supports GT2 and GT3, or GT2 and GT4.
      - *(last checked in GT 4.0.4)*
  - Verification is “simple”
    - Check if the certificate is a proxy. If so:
      - *Check the critical extensions.*
        - Check if the ProxyCertInfo extension is respected.
      - *Check if the certificate was signed by the previous one in the chain.*
      - *Usual verification procedure.*
    - If it is not a proxy, call the standard verification routine.

GSI/SSL Differences and Issues

- **Issue 3: Delegation**
  - If you do not need it, you're ok.
  - If you need, it is a problem.
    - The protocol is quite complex (SSL3_RT_GSSAPI_OPENSSL)
    - A mix between normal challenge/response, SSL handshaking, and certificate signing.
    - Globus said we can take its code if we need it.
  - VOMS does not need delegation.
    - Delegation is not supported

GSI/SSL Differences and Issues

- **Issue 4: OpenSSL (C/C++)**
  - If you do not link against globus you must support the OpenSSL version in the system.
  - If you do link against globus, and VDT >= 1.10, you must support the OpenSSL version in the system.
  - Two successive versions of OpenSSL (different version numbers, not just letters) are always incompatible with each other.
  - Problems seen up to now include:
    - Functions which became macros.
    - Prototypes changed incompatibly.
    - Structure initialization rules changed.
  - Most you notice while compiling, a select few you only discover at runtime!
  - Most only get noticed on specific architectures!
  - Have to delve in changelogs to discover what changed.
    - Sometimes it is not apparent.

VOMS without Globus

- **Version 1.8 (Released)**
  - The VOMS server accepts pure SSL connections as well as GSI.
- **Version 1.9 (In development - soon)**
  - The VOMS clients will make SSL connections
    - Requires VOMS server >= 1.8
  - The VOMS clients no longer link against globus.
- **Version 2.0 (Next year)**
  - The VOMS Server will no longer link against globus.
  - It will accept both GSI and SSL connections.
  - Different than what was previously proposed!

VOMS Clients without Globus

- **Can still speak GSI!**
  - Though it will speak SSL.
  - Implemented using the callout mechanism in OpenSSL.
  - Use normal SSL_Read()/SSL_Write() calls.
  - Wrapped be the GSISocketClient class
  - Most difficult thing: replicate the selection process for the default locations of certificates, private keys, CAs, proxies...
- **Can still create proxies for GT2, GT3, GT4**
  - Never relied on globus for that anyway...
- **Can still authenticate and verify proxies for GT2, GT3, GT4**
  - Never relied on globus for this either.
- **The same is true for the Java APIs.**

Libraries Differences

- **SL4**
  - libvomsapi will link against the system OpenSSL.
  - libvomsapi_<flavor> will link against the specific Globus flavor of OpenSSL.
  - libvomsapi-nog will be the same as libvomsapi.
  - Developers should link against libvomsapi_<flavor> if linking against globus, and libvomsapi otherwise.
- **SL5**
  - libvomsapi, libvomsapi_<flavor>, and libvomsapi-nog will all link against system OpenSSL.
    - Globus from VDT1.10 links against system OpenSSL, not against its own system.
  - It follows that developers should link against libvomsapi.

Libraries Differences

- **You may notice there was no mention of libvomsapic.**
  - That's because you are not supposed to link to it.
  - libvomsapi offers both the C and C++ interfaces.
  - Since version 1.5, libvomsapic is just a rename of libvomsapi.
    - Was in the release notes.
  - In effect, even if you are linking to libvomsapic, you are already linking to libvomsapi instead.

API Differences

- **C/C++**
  - VOMS_RetrieveFromCtx() and its C++ counterpart can no longer be implemented.
    - It is impossible to retrieve credentials from a context without linking against Globus.
    - Two functions will be provided separately that re-implement this API, and that should be linked against globus.
    - The prototype for the C one will be identical.
    - The prototype for the C++ one will change (obviously)
- **Java**
  - org.globus.gsi.GlobusCredentials can no longer be referenced in the APIs. Use org.glite.voms.contact.UserCredentials instead.
    - Did you use those classes?
    - Basically, the VomsProxyInit class. (getVomsProxy())
    - Do you need an adapter class for GlobusCredentials?

API Differences

How do you want them packaged?