VOMS e GSI

Vincenzo Ciaschini

EMI Security Workshop

25-26/5/09

Proxies without Globus

- **Can I use proxies but not use Globus?**
  - YES!
- **How?**
  - Well, if you can afford breaking backwards compatibility, it is easy.
    - Remember that OpenSSL >= 0.9.8k supports proxies.
    - Just activate it and use plain SSL everywhere.
  - Otherwise, you must speak both SSL and GSI
    - Which can be done
      - *New clients speak SSL*
      - *New servers speak both SSL and GSI*

What is GSI?

- **GSI is *not* a protocol!**
- **It is a family of protocols.**
  - It works by encapsulating the usual SSL messages in some way.
    - Uses the send_token and get_token functions passed to do the job.
- **Luckily, practically everyone uses the same functions**
  - I.e.: write the message size and then send the message.
  - So the same solution works everywhere

How to deal with GSI?

- **So, what is the solution server-side?**
- **Intercept the messages from the socket BEFORE they reach the SSL code.**
- **Determine if they are SSL or GSI**
  - Hint: Unless you send single messages of size X 272M <= X <= 320M, GSI messages do not have a first byte >= 20, <= 23.
- **During read:**
  - Strip the size
  - Collect the whole message.
  - Send the message to the SSL routines.
- **During write:**
  - Just write the size before you write the message.

How to deal with GSI (2)

- **For a C example, see src/socklib/Server.cpp in the VOMS sources:**
  - Note: dirty hack. Assumes you use processes rather than threads. Should be better handled by rewriting as an interceptor BIO.

- **For Java I am told that it is equally possible. Not sure how, though.**

How to Deal with GSI (3)

- **Message type may be 26 (RT_GSSAPI_SSL)**
  - Just remove the wrapping. The data is the real message.
  - This should only happen during delegation...

How to deal with Proxies

- **For RFC proxies:**
  - If you depend on OpenSSL >= 0.9.8k just activate proxy support and you're done.
    - Note: SL4 does not use OpenSSL >= 0.9.8k!
  - If you do not depend on it then you have to write code to support proxies yourself.
    - Read src/sslutils/sslutils.c,src/socklib/Server.cpp and src/api/java/org/glite/voms/PKIVerifier.java to see how it is done.
- **For non-RFC proxies**
  - You should not use them!
  - Are there some parts of gLite that still do not support them?
  - Read the same sources as before to see how it is done.

How to deal with delegation

- **If your client is a GSS client, the first message after the handshake may be '0' or 'D'.**
  - '0' means : No delegation required.
  - 'D' means: Delegation required.

- **If you do not support delegation, just eat the message and you're done.**

How to deal with delegation

- **The current code inside GSI is deprecated by globus developers themselves!**
  - Source: phone conf with globus devs, september 2008.
  - I do not support it.
- **It is not excessively complex to add support, but...**
  - Shouldn't we get rid of it instead?
- **In general, delegation is best done through an external service.**

Conclusions

- If you only use RFC proxies and can discard backwards compatibility, it is easy.
- If you only use RFC proxies but backwards compatibility is important, it is a little more complicated, but most of the work is server-side only.
- If you use non-RFC proxies, there is a LOT of code to add for basic proxy verification.

- Delegation is a mess. Find some other way to do it!