Use of a hardware token for Grid authentication by the MICE data distribution framework

Henry Nebrensky
(HEP Group, College of Engineering, Design, and Physical Sciences, Brunel University, UK)
Janusz Martyniak
(High Energy Physics Group, Department of Physics, Imperial College London, UK)

The international Muon Ionization Cooling Experiment (MICE) is designed to demonstrate the principle of muon ionisation cooling for the first time, for application to a future Neutrino Factory or Muon Collider. The experiment is currently running at the ISIS synchrotron at the Rutherford Appleton Laboratory, UK. As presently envisaged, the programme is divided into three Steps: characterisation of the muon beams (complete), characterisation of the Cooling Channel and Absorbers (data-taking restarting in 2016), and Demonstration of Ionisation Cooling (2018).

Data acquisition, in particular, can involve 24/7 operation for an entire ISIS cycle (4-6 weeks) and so for moving data from the DAQ to the Grid a valid, VOMS-enabled, Grid proxy must be made available continuously over that time. Long-lifetime proxies and password-less certificates raise security concerns regarding their exposure, whereas requiring a particular certificate owner to log in and renew the proxy manually at half-day intervals (as the VOMS extensions are limited to 24hrs) for weeks on end is operationally unsustainable. The solution is to use a "robot certificate" (one that does not represent a particular person) stored on a hardware token (from which it is not possible to extract the "private key" part of the certificate, thus proxies can only be used on the very machine that has the hardware token plugged into it).

```
~ > pkcs15-init -E                                                    <- Wipe card
Using reader with a card: Feitian ePass2003 00 00
~ > pkcs15-init --create-pkcs15 --profile pkcs15+onepin --use-default-transport-key
                --pin **** --puk **** --label "MICE Data Mover Robot 3"
Using reader with a card: Feitian ePass2003 00 00                     <- Initialise card
~ > pkcs15-init --store-private-key Robot_GridClientTest.p12 --format pkcs12
                --auth-id 01                                          <- Write existing key and certificate from disk
Using reader with a card: Feitian ePass2003 00 00
error:23076071:PKCS12 routines:PKCS12_parse:mac verify failure
Please enter passphrase to unlock secret key:
Importing 3 certificates:
0: /C=UK/O=eScience/OU=Imperial/L=Physics/CN=MICERobot:GridClient
1: /C=UK/O=eScienceRoot/OU=Authority/CN=UK e-Science Root
2: /C=UK/O=eScienceCA/OU=Authority/CN=UK e-Science CA 2B
User PIN [User PIN] required. Please enter User PIN [User PIN]:
~ > pkcs15-tool --list-certificates
Using reader with a card: Feitian ePass2003 00 00
X.509 Certificate [MICERobot]                                         <- Confirm key and certificate are on token
Object Flags : [0x2], modifiable
Authority : no
Path : 3f0050153100
ID : 84d9fcd5a4e9a7408301c9dc7e908bd4040895e4                         <- The ID allows us to store and use different certificates for different tasks in one token
GUID : {4f53ac2b-fcc1-aa0c-f8da-73edbb04160}
Encoded serial : 02 03 00A3E7
X.509 Certificate [UK e-Science Root]
...
~ > ./mkproxy.bash --slot 1 --bits=1024                               <- This script (from NIKHEF) does the OpenSSL incantations to generate the "raw" proxy
                   --id 84d9fcd5a4e9a7408301c9dc7e908bd4040895e4 --debug   Thanks, NIKHEF!
~> voms-proxy-init --noregen --valid 24:00 --voms mice:/mice/Role=mvr
Contacting voms.gridpp.ac.uk:15001 [/C=UK/O=eScience/OU=Manchester/L=HEP/CN=voms.gridpp.ac.uk] "mice"...
Remote VOMS server contacted succesfully.                             <- Add VOMS extensions
...
```

Transfer proxy from SL7 host to SL6 UI

We chose the Feitian ePass2003 because it was both significantly cheaper and easier to actually purchase; however at the time there was no software support for the hardware. The Feitian ePass2003 is certified by NIST (FIPS 140-2 Level 3).

On Linux, the ePass2003 is supported through OpenSC since version 0.13 but this package is not included in SL6 and the corresponding EPEL repo only included 0.12.
Scientific Linux 7.1 does include OpenSC v.0.13.0-9, which ironically is the one minor release for which ePass2003 support was temporarily broken

Created a "hybrid" SL7.1 system by installing packages from Fedora Core 20:

```
engine_pkcs11-0.1.8-7.fc20
      libp11-0.2.8-5.fc20
    opensc-0.13.0-11.fc20
```

This server is isolated from the network and does nothing apart from providing our SL6 UI with a raw Globus proxy.

Raw proxy is regularly re-created and provided to the UI by a cron job, where another drives the VOMSification. There is monitoring both for proxy validity and for the presence of the token itself – it does seem to lose the USB connection every 1 or 2 months (kernel?).

Now in production use since Sep. 2015, generating separate proxies for both RAW and Reconstructed data.

We are now preparing a CentOS 7 test machine – aim is to both generate the raw proxy and VOMSify on the Grid UI.

References

Feitian Technologies Co. Ltd.: "ePass2003"
http://www.ftsafe.com/onlinestore/product?id=3
(retrieved 17:40 3rd October 2016)

NIKHEF: "Using an Aladdin eToken PRO to generate grid proxies"
https://wiki.nikhef.nl/grid/Using_an_Aladdin_eToken_PRO_to_generate_grid_proxies

J. Martyniak: "A Grid-based Batch Reconstruction Framework for MICE"
CHEP 2015

J. Martyniak and the Mice Collaboration: "MICE data handling on the Grid"
J. Phys. Conf. Ser. 513 032063 (2014)

The MICE Collaboration : "MICE Raw Data"
https://dx.doi.org/10.17633/rd.brunel.3179644.v1 (2016)

Specific thanks to Jan-Just Keijser and NIKHEF colleagues for help with OpenSSL, proxy generation, and the mkProxy script; and Jens Jensen (UK *e*Science CA) for help with the robot certificate.

The work described here was made possible by grants from the Department of Energy and National Science Foundation (USA), the Instituto Nazionale di Fisica Nucleare (Italy), the Science and Technology Facilities Council (UK), the European Community under the European Commission Framework Programme 7, the Japan Society for the Propmotion of Science and the Swiss National Science Foundation, in the framework of the SCOPES programme, whose support we gratefully acknowledge. We acknowledge the use of Grid computing resources deployed and operated by GridPP in the UK. We are also grateful to the staff of ISIS for the reliable operation of ISIS.