VOMS SNOOPER (steve jones)

The Small Print:

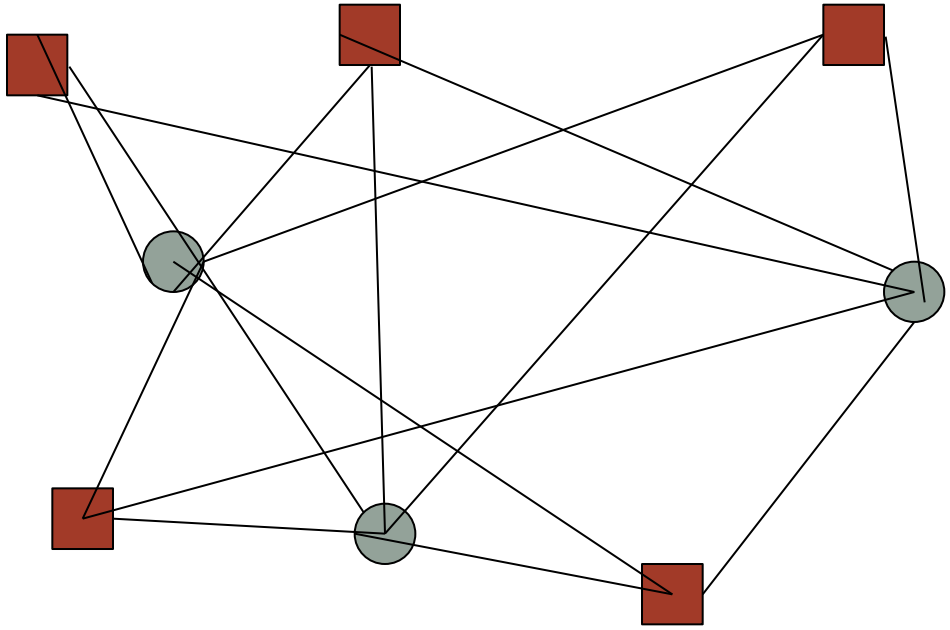
GridPP is part of the UK NGI. As such, we support related organisations outside LHC. The PMB has agreed that 1.0% of processing capability be allocated for non-LHC work. This is only a lower limit on what will be provided; all GridPP sites are encouraged to enable a defined list of VOs so that free CPU cycles are provided for the benefit of wider causes.

The configuration data for VOs is described in a set of tables containing the related YAIM variables. The variables that I am concerned with here are those pertaining to VOMS servers, used by a VO to validate users.

If these small VOs are to get their share, we need a way to manage their VOMs records.

History

Without Central DB: I wasn't there at the time, but this is clearly an area where some level of central coordination is needed due to the many-to-many between sites and VOs.

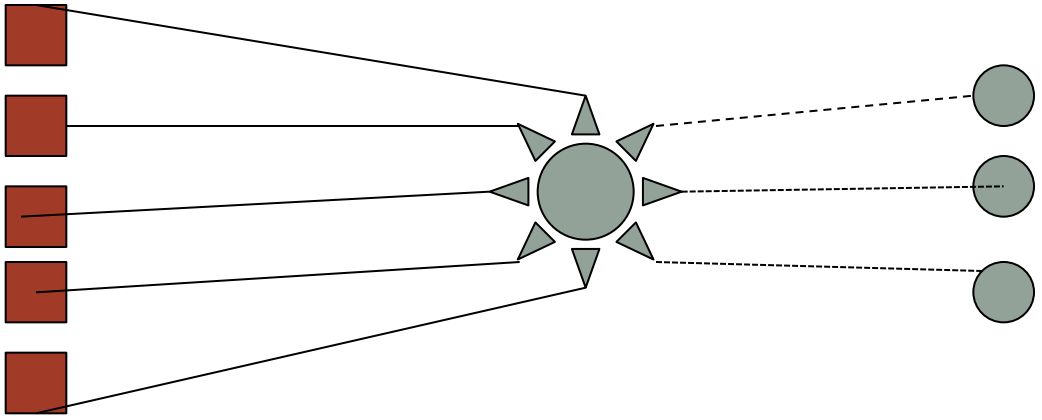


History …

With Central DB (referred to as the Operations Portal):

At http://operations-portal.egi.eu, the many to many between VOs and sites becomes a many to one between VO and DB, and another one to many between DB and site, i.e. the design works better.

The data is made available in the form of VOID cards, one for each VO. The cards are then consolidated into a single XML file which is available for download.



Missing Link?

The trouble is that there is no "standard" tool AFAIK to synchronise sites with the central DB. We need something to bridge that gap, i.e. read the XML VOID cards and write in the correct format. Question: what did sites do before?

Manual:

Pros: no expense, no training cost etc.
Cons: hard work, boring, repetitive, error prone, no standards, likely to be neglected, hard to spot errors, hard to read XML, etc. Basically unreliable (e.g. like host file maintenance, but harder).

Approved VOs Document:

Pros: A standard way to get at the data.

Cons: Prone to staleness. Poss. Incomplete snapshot. Many of the problems of manual update.

History …

UNIVERSITY OF LIVERPOOL

So write a tool to do this

That's what VomsSnooper was about. We needed a tool at Liverpool that could do this stuff for us.

I do "documentation" as a core task, and it would be sysiphusian to keep manually checking and editing the Approved VOs list.

But if I went to the trouble of scripting that, could other things be done too? To get at the requirements, we needed something simple that we could easily change.

It would have to be flexible enough to cover all (most?) of our needs so that the problem of site VOMS data management is swished away.

The approach was to coarsely carve a tool that could allow us to experiment with different requirements, making it general enough to suit many needs.

UNIVERSITY OF LIVERPOOL

Extra complication 1

SIDs or VODs?

Each VO has a set of records that are used by Yaim to configure a site. The Yaim variables can be given to Yaim in one of two formats. The original format, for inclusion within the site-info.def file itself, are known as SID records. Due to restrictions with DNS style names (with dots in them) it was later convenient to invent a new format, whereby the records are stored in their own separate files under a vo.d directory. I call this the VODs format. It is possible to represent any VO in VOD format. VOs with names with dots in them, however, can not be conveniently represented in SID format. VomsSnooper supports both representations, but VODs seem to be more straightforward, given the move to DNS style names. What should we use? What should new sites use? Traditional SIDs or modern VODs? Hint: VomsSnooper provides a tool to migrate from SIDs to VODs.

More complications

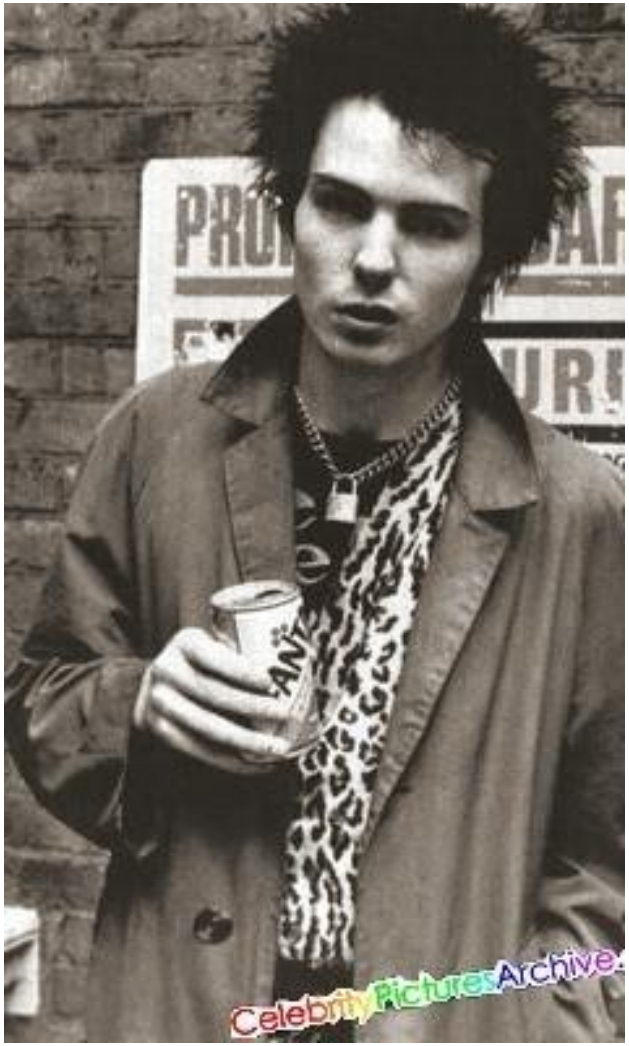
Silly SIDs (silly name for a sensible? idea):

Yaim variables can be given to Yaim in the site-info.def file are called (here) SIDs. The other type, for DNS names, are called VODs (because they live in the vo.d directory). If we print out DNS style VO names (e.g. na62.vo.gridpp.ac.uk) in SID format, they would look like this:

VO_NA62_VO_GRIDPP_AC_UK_VOMS_SERVERS

i.e. the idea is reaching the end of the road because it's getting hard to parse. Where does it start; the first VO_ or the second _VO_? It's messy, so I call these SILLY_SIDS. It's better to go to VODs at this point.

Sid



More complications …

- Perhaps some VOs "go it alone". I.e. they don't upload to the Operations Portal?

- Only certain variables are disseminated via the Operations Portal. Some VO variables not available in the Operations Portal are:
  VO_X_DEFAULT_SE
  VO_X_RBS
  VO_X_STORAGE_DIR
  VO_X_SW_DIR
  VO_X_VOMS_EXTRA_MAPS
  VO_X_VOMS_POOL_PATH

  These are the ones "acquired" at Liverpool. There may be more. How should these be disseminated?

One more complication …

I call this one the CERN rule, but I'm sure there is more than one rule at CERN. Anyway, this particular CERN rule relates to those VOs whose VOMS server reside at voms.cern.ch.

The VOMs records of those VOs should only contain data about voms.cern.ch. Data about other servers should be dropped, even if it exists in the VOID XML.

The relevant tools have a –ignorecernrule to turn off this behaviour if (when?) it becomes pest.

I don't know the provenance of the CERN rule; I only know it exists. I have a feeling it somehow relates to load balancing. Discuss?

Liverpool's Solution

VomsSnooper

Scope: Liverpool, but available for reuse.

Status: Functional testbed. It suits our needs, but is it general?

General functional requirements

- It must be easy to implement a SAX parser, as data from the Operations Portal comes in XML.
- It must support Objects as CIC data is "relational".

UNIVERSITY OF LIVERPOOL

Not this type of SAX

UNIVERSITY OF LIVERPOOL

Not this type of Object

Non-functional requirements

- Needs to be a command line tool to allow scripting (no clicking…)
- Needs to be mainstream language (easy to edit).
- Needs some form of change control etc.
- Needs some form of license I guess.
- We have no specific budget for development, so it must be SMART (Specific, Measurable, Attainable, Relevant, Time-bound), i.e. quick and easy. In this case, the work was worth it for Liverpool alone. Buy-in would improve that ratio.

Selection

- Perl - poor object support, easy but ugly.
- Python - learning curve (but I should have used it)
- Java - no learning curve, but "difficult", i.e. requires Eclipse.
- SVN, CVS, Git, etc.

Final choice:

Java (line of least resistance) for core tools, with perl, bash etc. for scripting use cases. GitHub to store the material. Academic Free License version 3.0?

Uses cases wanted at Liverpool

Name: fixApprovedVos

Description:

Get the newest records from the CIC portal XML and merge them into the wiki page of approved VOs, keeping it up to date.

Reason:

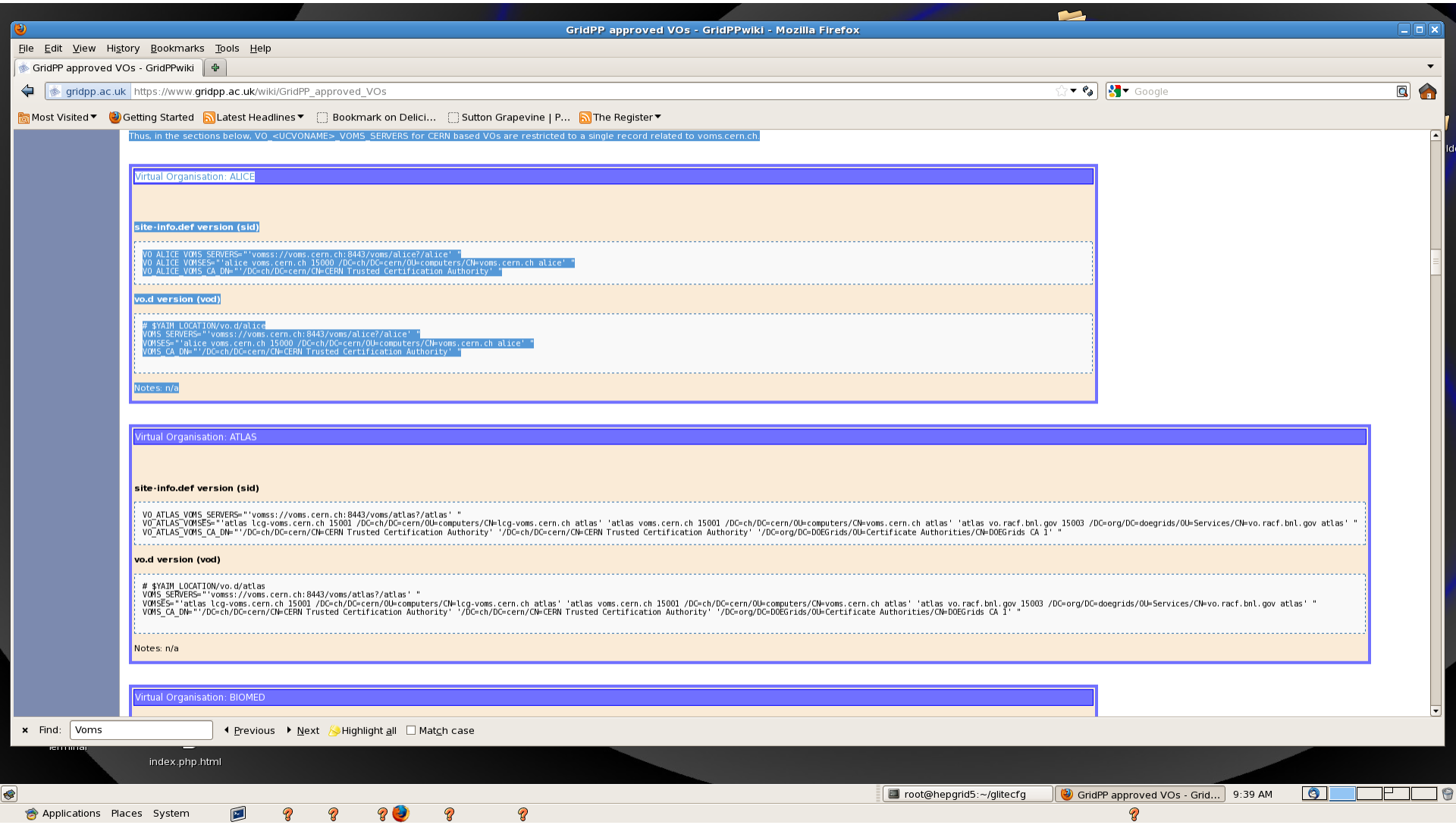
The Approved VOs list is in danger of getting stale. We need to automate the grunt work of that with this use case. We use the tool several times each month to keep the Approved VOs document up to date.

Priority:

Must have. It's already in use each week to maintain the restructured Approved VOs document at

https://www.gridpp.ac.uk/wiki/GridPP_approved_VOs

(aside: standard structure for Approved VOs file to allow automation)



UNIVERSITY OF LIVERPOOL

Uses cases …

Name: newVomsRecsForMySite

Description:

Download a new set of records for inclusion (by hand or automatically) into the local site-info.def file.

Reason:

When setting up a new site, maybe you just want to create a new set of VOMS records directly from the CIC portal, instead of using the Approved VOs list.

Also good for supporting VOs who do not appear in the approved VOs.

Priority:

Must have

UNIVERSITY OF LIVERPOOL

Uses cases …

Name: getLSCRecords

Description:

Generate new LSC records automatically from XML, suitable for inclusion in a /etc/grid-security/vomsdir directory, by-passing the usual yaim steps.

Reason:

Some sites may by-pass yaim. This creates the LSC files directly.

Priority:

Must have if you want to do this.

Uses cases …

Name: convertSidsToVods

Description:

Convert a site-info.def file that contains old, SID format VOMS records into the new style that uses discrete files under the vo.d directory. It does not use any of the main VomsSnooper components.

Reason:

It's a lot easier to manage this stuff if you have all SIDs or all VODs. Right now it's messy to compare.

Priority:

Must have if you need to do this

Uses cases …

Name: checkMyVomsRecords

Description:

Compare your set-up with the CIC portal XML and make sure they match well, or not, as the case may be.

Reason:

You need a way to check if your site is OK.

Priority:

Nice to have

Uses cases …

Name: checkMySite

Description:

Get the newest records from the CIC portal XML and make sure your current site-info.def/vo.d tallies with those records.

Reason:

You need a way to check it your site is OK. (I think this was a reimplementation of checkMySite!)

Priority:

Nice to have

Uses cases …

Name: checkMyLSCRecords

Description:

Compare the LSC records at a site to the ones that exists in the CIC Portal XML file.

Reason:

Sites may want to just compare the end result, irrespective of the site-info.def file.

Priority:

Nice to have.

Uses cases …

Name: listFQANs

Description:

List a VOs FQANs.

Reason:

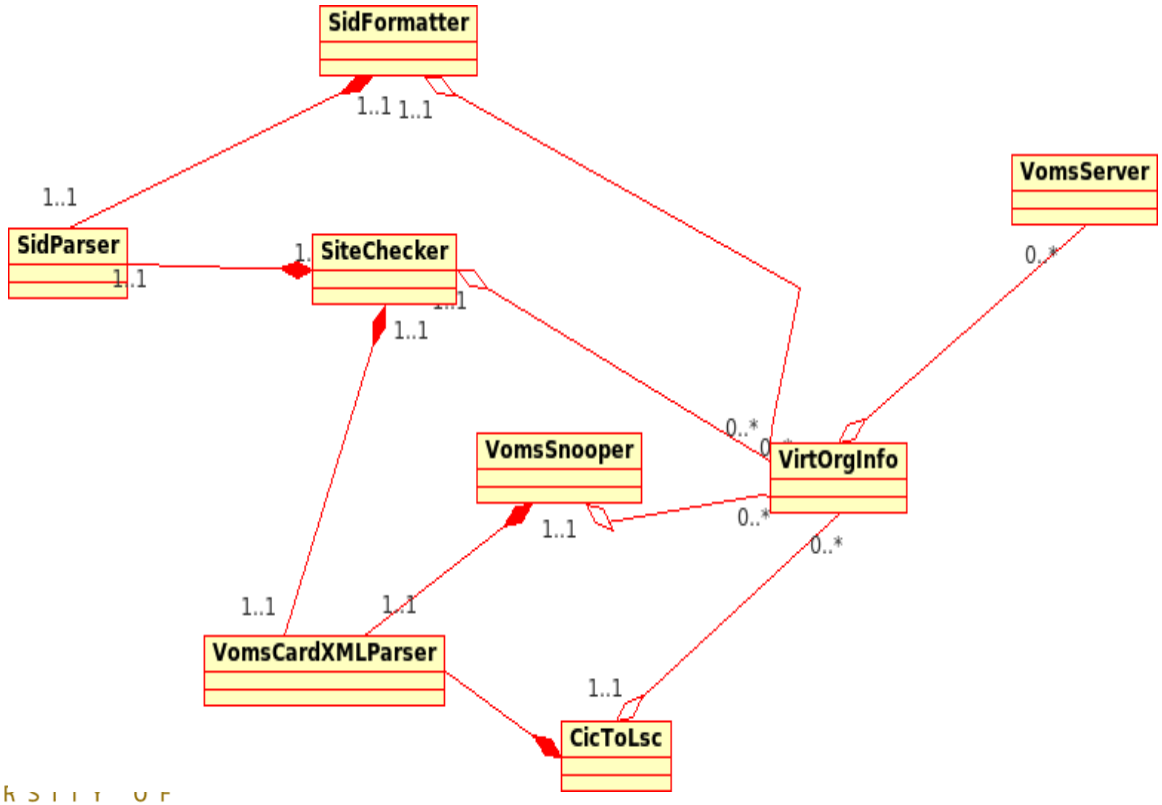
I thought it might be nice to list the FQANs; maybe I'll integrate it with users.conf/groups.conf sometime. It depends.

Priority: nice to have

Documents, Design …

Documentation: https://www.gridpp.ac.uk/wiki/VomsSnooper_Tools

Design:



UNIVERSITY OF LIVERPOOL

Install

Downloaded from : http://hep.ph.liv.ac.uk/~sjones
Document at: https://www.gridpp.ac.uk/wiki/VomsSnooper_Tools

The software is distributed as a tar file, that should be unpacked from your home directory. If you unpack or move it elsewhere, you'll have to update some paths to ensure the right files are found (see ~/git/GridDevel/vs_scripts/set_paths.sh). The tools are distributed as relocatable directly-executable JAR files that reside initially in ~/git/GridDevel/vs_scripts/ant/dist/lib.

Note: No build needs to be done, unless you edit the Java software. If you do need to build it, an ant makefile called build.xml is provided in the ~/git/GridDevel/vs_scripts/ant directory.

Options – VomsSnooper

The VomsSnooper tool takes an XML file from the CIC portal, and formats it into a standard, sorted manner for Yaim.

Mandatory arguments:

--xmlfile f # Input XML file downloaded from CIC portal
--myvos f # Names of VOs that I support
--vodfile f # Names of VOs that must be output in VOD format
--outfile f # Where to write SIDs (records that can be represented in a site-info.def)

Optional arguments:

--help # Prints this info
--voddir d # Where to write VODs (records that cannot be represented in a site-info.def)
--printvodtitle # When printing VODs, put the name of the VOD file in the output
--nosillysids # When printing SIDs, reject ones with silly names (DNS/ dot style)
--ignorecernrule # Ignore the cern rule (use only voms.cern.ch)
--extrafields # Print some extra fields (not recommended)
--contactsfile # Where to print the VO contacts (not recommended)
--fqans # File where to print VO FQANs
--vomsdir # Where to print LSC files

UNIVERSITY OF LIVERPOOL

Options – CicToLsc

The CicToLsc tool takes an XML file from the CIC portal, and creates a set of LSC files from the data, bypassing Yaim.

Mandatory arguments:
--xmlfile f # Input XML file downloaded from CIC portal
--approvedvos f # File of names of VOs that I support
--vomsdir # Where to print LSC Files
Optional arguments:
--help # Prints this info

Options – SidFormatter

The SidFormatter tool takes an existing populated site-info.def file and vo.d directory, and formats it into a standard, sorted manner. It writes its output into a new,
unpopulated site-info.def file and vo.d directory.

Mandatory arguments:
--oldsiddir dir # Some existing directory that contains
a populated site-info.def file and
vo.d directory
--newsiddir dir # Some existing directory that contains an
unpopulated site-info.def file and vo.d
directory
--myvos f # Names of VOs that I support
Optional arguments:
--flat # Print all the records out in a
site-info.def, even when they are
silly sids
--help # Prints this info

Options – SiteChecker

The SiteChecker tool checks if a site complies with the CIC portal XML

Mandatory arguments:
--xmlurl f # URL of XML file (i.e. CIC portal)
--sidfile f # Location of site-info.def file
Optional arguments:
--help # Prints this info

Too dry. Show how it works!

To install. Be in a terminal; cd $HOME

Linux:

```
# wget http://hep.ph.liv.ac.uk/~sjones/VomsSnooper-1.22.0.tar
# tar -xvf VomsSnooper-1.22.0.tar
# cd $HOME/git/GridDevel/vs_scripts
# . set_paths.sh
# VomsSnooper.jar -h
```

Workarounds for Mac OS X:

```
export
PATH=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/H
ome/bin:$PATH
java -jar
/Users/sjones/git/GridDevel/vs_scripts/ant/dist/lib/VomsSnooper.jar -h
```

Use cases under /Users/sjones/git/GridDevel/vs_scripts/usecases

UNIVERSITY OF LIVERPOOL

Fix Approved VOs

```
#!/bin/bash
source ../../set_paths.sh     # Set up the env
mkdir -p glitecfg/vo.d        # dir struct

# Get the XML (wget missing on mac)
curl -o VOIDCardInfo.xml http://operations-portal.egi.eu/xml/voIDCard/public/all/true
# Spit out the VODs and SIDs
java -jar /Users/sjones/git/GridDevel/vs_scripts/ant/dist/lib/VomsSnooper.jar --xmlfile
      VOIDCardInfo.xml  --myvos allvos.txt --vodfile allvos.txt --voddir glitecfg/vo.d --outfile
      glitecfg/site-info.def --nosillysids –printvodtitle
java -jar /Users/sjones/git/GridDevel/vs_scripts/ant/dist/lib/VomsSnooper.jar --xmlfile
      VOIDCardInfo.xml  --myvos allvos.txt --vodfile novos.txt  --voddir glitecfg/vo.d --outfile
      glitecfg/site-info.def --nosillysids --printvodtitle

# Merge it all into the original wiki
./assemble_content.pl -dir glitecfg/ -wf wiki.txt -of new.wiki.txt
```

New VOMS Records for my site

```
#!/bin/bash

source ../../set_paths.sh
mkdir -p void/deployed/vo.d; mkdir -p void/merged/vo.d
mkdir -p void/xml/vo.d

curl -o VOIDCardInfo.xml http://operations-portal.egi.eu/xml/voIDCard/public/all/true

cat  VOIDCardInfo.xml | sed -e "s%</VoDump>%%" > delme1
cat  ExtraVOIDCardInfo.xml | grep -v "^<VoDump>"  > delme2
cat delme1 delme2 > VOIDCardInfo.xml

java -jar /Users/sjones/git/GridDevel/vs_scripts/ant/dist/lib/VomsSnooper.jar --xmlfile
      VOIDCardInfo.xml  --myvos void/myvos.txt --vodfile void/vod.txt --voddir
      ~/git/GridDevel/vs_scripts/usecases/newVomsRecsForMySite/void/xml/vo.d --outfile
      /Users/sjones/git/GridDevel/vs_scripts/usecases/newVomsRecsForMySite/void/xml/site-
      info.def
```

UNIVERSITY OF LIVERPOOL

New VOMS Records for my site …

rsync -a --delete root@hepgrid6:/root/glitecfg/
$HOME/git/GridDevel/vs_scripts/usecases/newVomsRecsForMySite/void/deployed

./sid_merger.pl --oldsid void/deployed/site-info.def --deltas void/xml/site-info.def --newsid
void/merged/site-info.def

rsync -a --delete void/xml/vo.d/ void/merged/vo.d

I don’t like YAIM. Bring me the LSC Records!

```
#!/bin/bash

source ../../set_paths.sh
rm -rf vomsdir

rm  VOIDCardInfo.xml
mkdir -p vomsdir

curl -o VOIDCardInfo.xml http://operations-portal.egi.eu/xml/voIDCard/public/all/true
java -jar /Users/sjones/git/GridDevel/vs_scripts/ant/dist/lib/CicToLsc.jar --xmlfile
      VOIDCardInfo.xml  --approvedvos myvos.txt --vomsdir vomsdir

cat ./vomsdir/vo.sixt.cern.ch/voms.cern.ch.lsc
```

UNIVERSITY OF LIVERPOOL

I’ve had it with these SIDs - please convert them all to VODs.

rm vo.d/atlas
tar -xvf backup.tar
./convertSidsToVods.pl -sid site-info.def

Note: it will do this inline, so it is best to have a backup, just in case.

The other use cases

The other use cases are various scripts and tools for checking that the records on your site are correct.

They are all covered in the documentation.

I've tried to harden them against "dirty data", but it's hard to guard them against any type of input.

The use cases in this category are checkMyVomsRecords, checkMySite and checkMyLSCRecords.

Future enhancements

The question of VO expectations with respect to node-resident software came up last week.

VomsSnooper can be enhanced to list any of the fields in the Operations Portal XML.

A good example is the “listFQANs” use case, which can list out all the FQANs related to a spectic set of Vos. This could be useful for configuring (say) the users.conf/groups.conf files.

Discuss: should this be done for the node-resident software? If so, should the output be in the Approved VOs document in a standard format?

Discuss: is any other data relevant?

UNIVERSITY OF LIVERPOOL

Suggestions

New site:

- Go straight for VODs (no SIDs) with Yaim.
- Pull with VomsSnooper using newVomsRecordsForMySIte use case.
- Periodically re-run VomsSnooper with diff (or one of the “checker” usecases) to ensure continued integrity.

Old site:

- Convert to VODs only
- Treat as a new site. Use Yaim

Other Stuff

Try to get “generic” site-info.def for all servers, to minimize maintenance.

We need a way to manage those VO_ yaim variables that aren’t in the VOID cards, i.e. VO_X_DEFAULT_SE, VO_X_RBS, VO_X_STORAGE_DIR, VO_X_SW_DIR, VO_X_VOMS_EXTRA_MAPS, VO_X_VOMS_POOL_PATH.

Provenance of the CERN rule – what’s it all about?

Silly-sids – does Yaim cope (VO_NA62_VO_GRIDPP_AC_UK_VOMS_SERVERS)

Conclusion

Operation Portal – great idea, but a bridge to nowhere.

Hence, VomsSnooper.

Two places to pull VOMS Records: Approved VOs doc (maintained 2 x monthly currently by Mark Norman) or direct (e.g. VomsSnooper)

Other tools (slightly experimental) to check site layouts.