LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

GRIDKA SCHOOL 2012
KARLSRUHER INSTITUT FÜR TECHNOLOGIE
KARLSRUHE

August 30, 2012

dCache Introduction Course

Christoph Anton Mitterer
christoph.anton.mitterer@lmu.de

VIII. Examples And Exercises

Conventions

The following conventions are used:

- Lines starting with "`$`" are entered within a POSIX-`sh`-compatible shell.
- Lines starting with "`#`" are entered within a POSIX-`sh`-compatible shell, with the effective user-ID and group-ID being `0` ("root-rights").
- Lines starting with "`(`*`location`*`) >`" are entered within dCache's administration interface with *`location`* as the current location.
- Standard input is written black, standard output grey and standard error red.

5. Access Control

Devising A "Real World Scenario"

The following defines a "real world scenario" that will be used through the following exercises:

- All the users have unique grid certificates with distinct DNs.
- All users are members to the `dech`-VO (via their grid certificates).
- Some users have the `production`-role for that VO.
- There are two classes of files to be stored:
  - "normal" files
    These are "normal" files from the users, for example personal data.
  - "production" files
    These are files used with an important production-system (for example an automated service that is critical for the organisation).
- All users should be allowed to read all files.
- Only users that are a member of the `dech`-VO should be allowed to write to "normal" files and directories.
- Only users also having the `production`-role should be allowed to write to "production" files and directories.

E1: Enabling gPlazma And Basic Configuration

The goal of this exercise is to enable the `gplazma`-service and set up its basic configuration.

1. Verify that the following service is enabled in the "layout-configuration-file" on a node of the cluster:
   `[gPlazmaDomain]`
   `[gPlazmaDomain/gplazma]`
   Of course, any domain can be used.
2. Verify that the CA- and VOMS-root-certificates, that should be used and trusted, are present and configured in `/etc/grid-security/certificates/` and `/etc/grid-security/vomsdir` respectively.
3. The other general gPlazma configuration-options need not to be touched.
4. In this and the following exercises, a combination of the `x509`, `voms`, `vorolemap`, and `authzdb` plug-ins shall be used, as described in chapter V.

E1: Enabling gPlazma And Basic Configuration

5. Configure the plug-ins in the gPlazma plug-in configuration file (per default found at `/etc/dcache/gplazma.conf`):

```
auth    optional        x509
auth    optional        voms

map     requisite       vorolemap
map     requisite       authzdb

session requisite       authzdb
```

   Remove any other (especially the default) entries:

```
auth    requisite       gplazma1
map     requisite       gplazma1
session requisite       gplazma1
```

6. Try to understand this configuration.
7. Other plug-in-specific configuration-options in the dCache configuration files need not to be touched.
8. Restart the `gplazma`-service:

```
# dcache restart gPlazmaDomain
```

E2: Creating The "grid-vorolemap-File"

The goal of this exercise is to create a "grid-vorolemap-file" with mappings that fit to the "real world scenario".

1. Create the file `/etc/grid-security/grid-vorolemap` with the following contents:

```
"*" "/dech" dech_normal
"*" "/dech/Role=production" dech_production
```

This has the following semantic meanings:
Certificates with any DN and the FQAN "`/dech`" will be mapped to the virtual user-name `dech_normal`.
Certificates with any DN and the FQAN "`/dech/Role=production`" will be mapped to the virtual user-name `dech_production`.
If a certificate has both FQANs attached it is mapped to both virtual user-names.

E3: Creating The "storage-authzdb-File"

The goal of this exercise is to create a "storage-authzdb-file" with mappings that fit to the "real world scenario".

1. Create the file `/etc/grid-security/storage-authzdb` with the following contents:

```
version 2.1

authorize dech_normal read-write 1000 1000 / / /
authorize dech_production read-write 1001 1000 / / /
```

This has the following semantic meanings:

- The virtual user-name `dech_normal` is mapped to the actual UNIX user-ID `1000` ("`dech`") and the actual UNIX group-ID `1000` ("`dech`").
- The virtual user-name `dech_production` is mapped to the actual UNIX user-ID `1001` ("`dech_prod`") and the actual UNIX group-ID `1000` ("`dech`").
- For both, the allowed access-mode is not generally restricted to read-only.

E4: Generating VOMS Proxy Certificates

The goal of this exercise is to show how to generate VOMS proxy certificates.

1. Optionally, read the documentation to the `voms-proxy-init`, `voms-proxy-destroy` and `voms-proxy-info` programs.
2. Create a plain proxy certificate with membership to the `dech`-VO via:
   ```
   $ voms-proxy-init -voms dech:/dech
   ```
   Here, "`dech`" specifies the VOMS-server to be used, while "`/dech`" specifies the FQAN to be requested.

   The following is more or less equivalent (it actually lets the server decide which FQANs he assigns) but shorter:
   ```
   $ voms-proxy-init -voms dech
   ```
3. Analyse that proxy certificate via:
   ```
   $ voms-proxy-info --all
   ```
4. Optionally, destroy the proxy certificate via:
   ```
   $ voms-proxy-destroy
   ```

E4: Generating VOMS Proxy Certificates

5. Create a proxy certificate with membership to the `dech`-VO and the `production`-role via:
   ```
   $ voms-proxy-init -voms dech:/dech/Role=production
   ```
   When analysing that proxy certificate it should be noted, that it got both FQANs attached by the VOMS-server:
   ```
   attribute : /dech/Role=production/Capability=NULL
   attribute : /dech/Role=NULL/Capability=NULL
   ```
   It should also be noted, that the attached FQANs have an order, which might have an influence at further actions using that proxy certificate (the order can be changed via `voms-proxy-init`'s `-order`-parameter).
6. Optionally, create a proxy certificate with membership to the `dech`-VO and multiple roles via:
   ```
   $ voms-proxy-init -voms dech:/dech/Role=production -voms dech:/dech/Role=GKS
   ```
7. Destroy the proxy certificate.

E5: Checking The Mappings

The goal of this exercise is to see how different combinations of DNs and FQANs are mapped to different actual UNIX user-IDs and group-IDs.

1. Log in to dCache's administration interface and connect to the `gPlazma`-cell via:
   ```
   (local) > cd gPlazma
   ```
2. Check the mappings for different combinations of DNs and FQANs via:
   ```
   (gPlazma) > test login principal [principal]*
   ```
   For example:
   ```
   (gPlazma) > test login "dn:/C=DE/O=GermanGrid/OU=dech-school/CN=gs001" "fqan:/dech"
   Login[dech_normal,1000:[1000],[ReadWrite, HomeDirectory[/], RootDirectory[/]]]
   (gPlazma) > test login "dn:/C=DE/O=GermanGrid/OU=dech-school/CN=gs001"
   "fqan:/dech/Role=production"
   Login[dech_production,1001:[1000],[ReadWrite, HomeDirectory[/], RootDirectory[/]]]
   (gPlazma) > test login "dn:/C=DE/O=GermanGrid/OU=dech-school/CN=gs001"
   CacheException(rc=10018;msg=login failed)
   ```
3. It should be noted, that mappings without an FQAN fail.
   This is desired, as the "grid-vorolemap-file" set up before does not map single DNs ("explicit-DN-matches").

   Actually, any DN with one of the matching FQANs will lead to a mapping.

E5: Checking The Mappings

4. Try out the "`explain login`" command, which takes the same arguments than "`test login`" but gives more detailed output on how gPlazma's plug-ins decide.

Bugs Affecting E6

Unfortunately we currently have to deal with a few bugs:

- A bug in either dCache or the Linux kernel, which prevents `chmod` from working on NFS 4.1 mounts, affects E6.
  Switch to NFS 3 as described in chapter 3 (“Installation”). Do not forget to un-mount `/pnfs` and mount it again as NFS 3!
  This can be done for example via:
  ```
  # umount /pnfs
  # mount -o nfsvers=3 localhost:/pnfs /pnfs
  ```
- A very old bug, which makes dCache ignoring some FQANs of a proxy certificate with multiple FQANs, affects E6!
  In order to get expected results, proxy certificates should be created with only one FQAN.

E6: Trying Out Access Control

The goal of this exercise is to actually try out access control by writing files to the cluster.

1. Create two directories that will be used as "write-areas" via:
   ```
   # mkdir -p /pnfs/dcache.org/data/exp1/normal
   # mkdir -p /pnfs/dcache.org/data/exp1/production
   ```
   These directories were created with the traditional POSIX file permission modes `rwxr-xr-x` and are thus readable by anyone but writeable only by their respective owning users.
2. Set the owning user and group so that one of the directories is owned by the "normal user" and the other one by the "production user" via:
   ```
   # chown 1000:1000 /pnfs/dcache.org/data/exp1/normal
   # chown 1001:1000 /pnfs/dcache.org/data/exp1/production
   ```
   The user-IDs and group-IDs used above correspond to those that were set in the "storage-authzdb-file".
3. Understand, that it is not necessary at all for dCache, that UNIX users and groups with the corresponding IDs exist. It is however useful for many POSIX programs.

E6: Trying Out Access Control

4. Repeat the following with different VOMS proxy certificates and for different "write-areas":
   1. Create a VOMS proxy certificate (as described in E4) with zero or more FQANs.
   2. With an GSI-secured client, try to upload a file into one of the "write-areas", for example via one of the following:

      Using gsiDCAP, where the pool-node initiates the data-connection to the client-node:

      ```
      $ dccp /bin/sh gsidcap://$(hostname -f)/pnfs/dcache.org/data/exp1/normal/test1
      ```

      Using gsiDCAP, where the client-node initiates the data-connection to the pool-node:

      ```
      $ dccp -A /bin/sh gsidcap://$(hostname -f)/pnfs/dcache.org/data/exp1/normal/test2
      ```

      Using GridFTP, where the data flows via the door-node ("GridFTP version 1"):

      ```
      $ globus-url-copy file:///bin/sh gsiftp://$(hostname -f)/pnfs/dcache.org/data/exp1/
      normal/test3
      ```

   You should see some successful and some denied transfers, depending on your selected proxy certificate and "write-area".

Finis coronat opus.