Using GSI Authentication in Condor

Hao Wang

January 24, 2003

1 Introduction

This document describe how to set up Condor to use Globus' GSI authentication mechanism. The document is written using following configurations: Globus Tool Kit 2.0, Condor Version 6.5. In addition, we assume the followings to be true:

- There is an acceptable X.509 certificate to be used by Condor. The certificate can be either a host certificate or user certificate
- Every user who plans to submit jobs through Condor must also have an acceptable X.509 certificate
- The certificate(s) of the CA(s) who signs the certificates in 1 and 2 are available
- The Condor systems has been installed (but not necessarily configured)

2 Configuration

There are several different ways to set up Condor. This section first describes configuration changes that are common to all setups. Then specific requirements are described according to different environment setups.

2.1 General Configuration Parameters

Following configurations are common to all setups.

- `SEC_[LEVEL]_AUTHENTICATION = REQUIRED`
  This parameter turns on authentication in Condor. `LEVEL` can be one of the following: `CLIENT, SERVER, ADMINISTRATOR, DEFAULT`. For most purpose, use `DEFAULT`. For example: `SEC_DEFAULT_AUTHENTICATION = REQUIRED`
- `SEC_[LEVEL]_AUTHENTICATION_METHODS = GSI`
  This parameter set the default authentication mechanism(s) to use. Mutliple authentication mechanisms can be specified, separated by a comma. `LEVEL` can be one of the following: `CLIENT, SERVER, ADMINISTRATOR, DEFAULT` For most purpose, use `DEFAULT`. For example: `SEC_DEFAULT_AUTHENTICATION_METHODS = GSI`
- `SEC_[LEVEL]_ENCRYPTION = REQUIRED`
  This method turns on the encryption requirement. `LEVEL` can be one of the following: `CLIENT, SERVER, ADMINISTRATOR, DEFAULT`. For example: `SEC_DEFAULT_ENCRYPTION = REQUIRED`

- SEC_[LEVEL]_CRYPTO_METHODS = <ENCRYPTION METHOD>*
  This method sets the default encryption mechanism. LEVEL can be one of the following: CLIENT, SERVER, ADMINISTRATOR, DEFAULT. ENCRYPTION METHOD can be one of the followings: BLOWFISH, 3DES. For example: SEC_DEFAULT_CRYPTO_METHODS = 3DES

- SEC_[LEVEL]_INTEGRITY = REQUIRED
  This method turns on the integrity check requirement. LEVEL can be one of the following: CLIENT, SERVER, ADMINISTRATOR, DEFAULT. For example: SEC_DEFAULT_ENCRYPTION = REQUIRED

  NOTE: Current release of Condor uses MD5 as the default integrity check algorithm. Future release of Condor will allow Condor users to select preferred integrity check algorithms from a list of supported mechanisms.

- GRIDMAP=<location where GRID map file is installed>
  This parameter tells Condor where to locate the Grid Map file. The map file is very similar to Globus' grid map file. The difference is that it maps from Distinguished Name (DN) to userid@domain (A sample of the map file can be found in the Appendix section of the document.) For example,
  GRIDMAP=/usr/local/Condor/grid-mapfile

2.2 Specific Configuration Parameters

Following parameters use different settings based on the type of certificates (user certificate vs. host certificate) Condor is configured to use.

- GSI_DAEMON_NAME = [DAEMON_NAME_FORMAT]*
  This parameter tells Condor clients the possible name(s) of the server it should be expecting to contact during GSI authentication. DAEMON_NAME_FORMAT is of this format: /C=?/O=?/O=?/OU=?/CN=?. For example,
  GSI_DAEMON_NAME=/C=US/O=Condor/O=University of Wisconsin/OU=
  Computer Sciences Department/CN=condor/$$(FULL_HOST_NAME)

  Generally speaking, if host certificates are to be used by Condor, then the host's DN should be used. If a user certificate is to be used by Condor, then the user's DN should be used. More specific examples are given at later sections.

  Note the macro FULL_HOST_NAME. This macro is designed to facilitate the configuration of a large number of Condor hosts to use host certificate. If your organization decides to have a host certificate with the same name for Condor to use–in the example above, it is condor–then you can simply specify condor/$$(FULL_HOST_NAME) and this macro will get expanded at run time to match the host which the client is authenticating with.

  Multiple daemon names may be specified, each separated by a comma. For example,
  GSI_DAEMON_NAME=/C=US/O=Condor/O=University of Wisconsin/OU=
  Computer Sciences Department/CN=condor, /C=US/O=Condor/O=University of Wisconsin/
  OU=Computer Sciences Department/CN=alice
  In this case, during authentication, the client will make sure the server it is contacting matches one of the specified daemons, starting from the leftmost (condor).

- Map user/host DN name The Distinguished Names (DNs) can be rather long and tedious to use. Therefore, Condor allows the administrator to setup a Map file to map from DNs to local UIDs. This map file is very similar to Globus' Grid Map file. The only difference is that Condor requires the UIDs to be in the form: `id@domain` instead of `id`. For example, `alice@cs.wisc.edu` is a valid UID name. The system administrator must create a Map file for condor and add all users/hosts who are permitted to use Condor. Details of the map file is provided in the Appendix.

- Give authorization permissions Condor also provide user and host based authorization. Therefore, authorization policy may be specified to grant/deny users and hosts various access permissions. This document only covers the syntax of the authorization statement. For details on how authorization works in Condor, please refer to the Condor manual. The authorization parameter's format is as follows:
  `HOST[ACTION]_[LEVEL] = id@domain/host`
  Here `ACTION` is either `ALLOW` or `DENY` and `LEVEL` can be `READ, WRITE, OWNER, ADMINISTRATOR. NEGOTIATOR_SCHEDD, and DAEMON`. For the purpose of this document, we only consider two authorization levels: `READ` and `WRITE`. `id@domain` is the UID mapping used in the map file. `host` can either be a specific host (`b05.cs.wisc.edu` for example) or a subnet (`*.cs.wisc.edu` for example).

  At this point, we assume that there exists a Condor map file that maps from users/hosts DNs to UIDs. Now, suppose that users `alice@cs.wisc.edu` and `bob@cs.wisc.edu` (both are mapped UIDs) are given permissions to submit jobs using Condor. Then following authorization parameters must also be set up.

  ```
  HOSTALLOW_READ = */*.cs.wisc.edu
  HOSTALLOW_WRITE = alice@cs.wisc.edu/*.cs.wisc.edu, bob@cs.wisc.edu/*.cs.wisc.edu
  ```

  The first statement allows both users to perform tasks such as check pool status and job status. The second statement allows both to submit jobs (which require write access permission) to the pool.

  Now, let us assume that Condor daemons are running on the host `condor.cs.wisc.edu` and use a host certificate and the DN is mapped to the UID `condor@cs.wisc.edu`. Then we also need to grant additional permissions to the UID `condor@cs.wisc.edu`:

  ```
  HOSTALLOW_WRITE = alice@cs.wisc.edu/*.cs.wisc.edu, bob@cs.wisc.edu/*.cs.wisc.edu,
  condor@cs.wisc.edu/*.cs.wisc.edu
  ALLOW_DAEMON = condor@cs.wisc.edu/*.cs.wisc.edu
  ```

2.3 Configuring Condor to run as normal user (no root privilege)

In this configuration, Condor will run as a normal user process without special root privilege. There are two possible ways to configure Condor in this situation, depending on what kind of certificate you have for Condor.

2.3.1 Using user proxy

In case you want to let Condor to use the proxy file created by the user, you need to do the following.

1. Set up the common configurations listed in section 2.1
2. Configure the authorization policy (section 2.2
3. Run grid-proxy-init to create a proxy
4. Start Condor daemons and try to submit a job

2.3.2 Using user certificate

In case you want to let Condor to use the user certificate file directly, you need to do the following.

1. Make sure that the `userkey.pem` file is not encrypted. Normally, the `userkey.pem` file is encrypted for security reasons and user needs to run `grid-proxy-init` or similar tools to obtain a short term proxy file to use. By storing the key file in clear-text, you may run into the risk of losing the key. To convert an encrypted private userkey into a decoded format, please consult the `rsa` command in OpenSSL.
2. Set up the common configurations listed in section 2.1
3. Set up user certificate environment variable
   In Condor configuration file, set the parameter `GSI_DAEMON_DIRECTORY` to point to the location where the user certificate is installed. For example,
   $GSI\_DAEMON\_DIRECTORY = /home/alice/CondorUserCert$

   NOTE: by setting this variable, Condor will set the following environment variables:
   ```
   X509_DIRECTORY = $GSI_DAEMON_DIRECTORY
   X509_USER_CERT = $GSI_DAEMON_DIRECTORY/usercert.pem
   X509_USER_KEY = $GSI_DAEMON_DIRECTORY/userkey.pem
   X509_CERT_DIR = $GSI_DAEMON_DIRECTORY/certificates
   ```
4. Configure the authorization policy (section 2.2
5. Run grid-proxy-init to create a proxy
6. Start Condor daemons and try to submit a job

2.4 Configuring Condor to run as root

In this configuration, Condor will run as a root process. In this situation, a host certificate is normally required.

1. Set up the common configurations listed in section 2.1
2. Set up host certificate environment variable
   In Condor configuration file, set the parameter `GSI_DAEMON_DIRECTORY` to point to the location where the user certificate is installed.

   You may skip setting this variable if the host certificate is located in the default Globus location: `/etc/grid-security`.

For example,
```
GSI_DAEMON_DIRECTORY = /home/condor/CondorHostCert
```

NOTE: by setting this variable, Condor will set the following environment variables:
```
X509_DIRECTORY = $GSI_DAEMON_DIRECTORY
X509_USER_CERT = $GSI_DAEMON_DIRECTORY/hostcert.pem
X509_USER_KEY = $GSI_DAEMON_DIRECTORY/hostkey.pem
X509_CERT_DIR = $GSI_DAEMON_DIRECTORY/certificates
```

3. Configure the authorization policy (section 2.2

4. Add the host DN to Condor's grid-map file. Since Condor's daemons also needs to authenticate to each other, the host's DN must also be included in the map file. An example of the map file is included in Appendix 2.4

5. Run grid-proxy-init to create a proxy

6. Start Condor daemons and try to submit a job

A Condor Grid Map file

The Condor Grid Map file is used to map users from their X.509 Distinguished Names (DNs) to local uids. The map file is very similar to Globus' GridMap file. The only difference is that Condor's map file maps to the format `uid@domain` instead of `uid` only, as in Globus' GridMap file. Following is an example of the Condor Grid Map file.

"/C=US/O=Condor/O=University of Wisconsin/OU=Computer Sciences Department/CN =condor_daemon/middle.cs.wisc.edu" condor_daemon@cs.wisc.edu

"/C=US/O=Condor/O=University of Wisconsin/OU=Computer Sciences Department/CN= John Doe" john@cs.wisc.edu

The first line represents a host DN mapping and the second line represents a user DN mapping.

B Sample Condor Configuration file (partial)

Authorization policy
HOSTALLOW_READ = */*.cs.wisc.edu
HOSTALLOW_WRITE = */*.cs.wisc.edu
Turn on GSI authentication
SEC_DEFAULT_AUTHENTICATION_METHODS = GSI
SEC_DEFAULT_AUTHENTICATION = REQUIRED
Turn on encryption
SEC_DEFAULT_CRYPTO_METHODS = 3DES, BLOWFISH
SEC_DEFAULT_ENCRYPTION = REQUIRED
Turn on integrity check
SEC_DEFAULT_INTEGRITY = REQUIRED
Where is the GRIDMAP file
GRIDMAP=/usr/local/Condor/condor_ca/grid-mapfile
Daemon's certificate directory
GSI_DAEMON_DIRECTORY=/usr/local/condor/CondorHostCert
Daemon's Distinguished Name
GSI_DAEMON_NAME =/C=US/O=Condor/O=University of Wisconsin/OU= Computer Sciences Department/CN=condor-daemon/$$(FULL_HOST_NAME)