SAM LANG AND SAM MEDER

Security and Credential Management on the Grid

Certificates and Globus Toolkit Security

The Grid extends the concept of a cluster to a more heterogeneous environment, where control and management of different clusters and systems are often the responsibility of entirely different organizations. The standard security mechanisms for clusters (such as standard Unix accounts, passwords, and other administrator controls) do not scale to environments managed by different organizations. Thus, security mechanisms must be devised that still provide the appropriate levels of access and control but can be managed in a distributed manner.

The *Globus Toolkit*™ provides software for secure communication, authentication, and authorization within the context of Grid computing. Central to this technology is the notion of a *PKI (public key infrastructure)* credential. A PKI credential consists of a certificate and its corresponding private key. These credentials allow systems to be secured on an interorganization basis and are required for both users and resources on a Grid supporting Globus Toolkit security. In PKI terms, users and resources are considered end entities. Certificates associated with these entities are appropriately called *end-entity certificates* or *EECs* and make up a majority of the certificates involved in setting up a PKI deployment.

In this column we will focus on the certificate part of a PKI credential. A *certificate* consists of three parts:

- A set of metadata, such as the identity of the owner of the key pair and the lifetime of the key pair
- The public key corresponding to the private key part of the credential
- A digital signature generated by using the private key of the issuer. This signature binds the metadata to the public key

The issuer of certificates is called a *CA (certificate authority)*. The CA verifies the identity of the person requesting the certificate and assigns a appropriate identity to the certificate. This verification can vary from setup to setup. Some certificate authorities may, for example, require requestors to apply to the CA in person and conduct extensive background checks, whereas others will simply issue certificates based purely on the requestor-supplied content of a simple web form.

Thus far, we have described a fairly generic PKI. The Globus Toolkit differs from other the generic model in that it introduces the *proxy certificate*. A proxy certificate (hereinafter *proxy*) is a certificate generated from an end-entity certificate and acts as a representative of the identity of said certificate. In other words, it acts as a proxy for the issuing certificate, a concept which is similar to that of proxies in the context of voting. There are several advantages to this:

- One no longer needs to use the private key associated with the certificate. The key of the EEC is generally stored in a more secure fashion (e.g., password-protected storage) so automated retrieval becomes difficult. In other words, using proxy certificates provides single sign-on capability.
- The lifetime of the proxy can be kept short. This reduces the exposure should the private key associated with the proxy certificate be compromised.
- Proxy certificates provide a mechanism for creating certificates that contain only a subset of the rights granted to the identity represented by the certificate.

We note that the proxy certificate technology is about to become a RFC in the IETF PKIX working group (www.ietf.org/internet-drafts/draft-ietf-pkix-proxy-08.txt).

How to Acquire a Certificate

Certificates, as mentioned above, are required for the operation of the software provided by the Globus Toolkit. This requirment leads to the question of how to find a provider of certificates that meets the security and operational needs you may have. As a first step, you should determine what certificate authorities are available. In general, you have three options for obtaining certificates: noncommercial CAs, commercial CAs, and do-it-yourself CAs.

A multitude of noncommercial CAs available. These generally fall into the categories of being very simplistic, for example, the Black Helicopter CA (jis.mit.edu/bh/getone.html), or CAs associated with a specific organization, for example, the DOE Grids CA (https://pki1.doegrids.org), which often

require membership in the organization.

Commercial CAs, such as Verisign (www.verisign.com), are also a possibility but may not be viable for economic reasons, especially when the number of resources and people involved is large.

The third option, setting up your own CA, may be suitable if you either need a certificate authority with basic capabilities, in which case the SimpleCA product described later in this article may meet your requirements, or if you require a more secure CA and have the in-house expertise required for running a CA, in which case you can use something like the software provided by OpenCA (www.openca.org).

Before picking a CA, you should carefully evaluate any security, interoperability, and potential legal requirements you have.

Using the Globus Simple CA

For Globus Toolkit users who can't get a certificate from another certificate authority, usually the best option is to install the Globus Simple CA package, which allows you to setup your own CA (option 3 from the previous section) and generate certificates yourself (including your own CA certificate), without the dependency on a CA elsewhere. It was designed to make CA management relatively painless for the beginning user.

Step 1: Installation

The first step is installation of the Globus Simple CA package. This will install into your Globus installation tools and utilities needed in future steps. The Globus Simple CA package is provided as a *GPT (Grid Packaging Toolkit)* package (much like *RPM*), making installation easy. The installation command to use is

```
# gpt-build globus_simple_
  ca_bundle-latest.tar.gz
  \<flavor type\>
```

The `\<flavor type\>` is used by the Globus Toolkit to specify platform- and compiler-specific information. In this case, the `\<flavor type\>` should be the same as the rest of the Globus Toolkit installation. See the Resources for more information about Globus and the Globus Simple CA.

Step 2: CA Setup

The primary purpose of the second step is to generate the CA certificate. We will also generate configuration files needed to use the CA with the Globus Toolkit. This is done by using a setup script located within the Globus Distribution at

```
# $GLOBUS_LOCATION/setup/
globus/setup-simple-ca
```

This script is automatically run by the *gpt-postinstall* command, so there is no need to run it by hand. The script will prompt you for information about the CA you wish to create. Most of the questions provide default answers, so if you are unsure, it's best to use the default. Here are the pieces of information you'll need to gather before running the setup script:

- *Subject Name of the CA*: The subject name of the CA uniquely identifies it among other Grid CA certificates. An example subject name is

```
cn=Sams CA,
ou=SamsLittleGrid, o=Grid
```

In this example, the organization is `Grid`, which uniquely identifies this CA as one that generates certificates used on the Grid. The organizational unit is `SamsLittleGrid`, which identifies this CA from other Grid CAs, and the common name is `Sams CA`, which identifies this particular certificate as the CA certificate within the *SamsLittleGrid* domain.

- *Email* - This should be the email address where you intend to receive certificate requests.

- *Expiration* - This is the number of days for which the CA certificate is valid. Once this time expires, the CA certificate will have to be recreated, and all of its certificates regranted.

- *Passphrase* - The passpharse of the CA certificate will be used only when signing certificates (with grid-cert-sign). It should be hard

What should I do with the generated CA setup package?

Anyone you can convince that you are a trustworthy CA (good luck!) should install the package. In reality, you will want to install it on all the machines that you want to include as part of your grid. The choice of distribution is up to you.

How many CAs should I set up for a Grid?

The entire Grid should use the same CA, so you should install the Globus Simple CA and set up your own CA (as described in Step 2) only once.

to guess, as its compromise may compromise all the certificates signed by the CA.

The end results of running the setup script are the CA certificate and configuration files, which are packaged up in a GPT distribution package, to be installed on other systems. The filename of the generated GPT package in our example would be: `globus_simple_ca_177ff0ce_setup-0.12.tar.gz`.

The filenames (and package) are encoded with the hash of the subject name of the CA. For example, the CA subject mentioned above is given the hash `177ff0ce`, so the CA certificate is given the filename `177ff0ce.0`. The hash of the CA certificate uniquely identifies this CA from others (a Globus Toolkit installation is allowed to trust multiple certificate authorities).

Step 3: Creating a Certificate

Once you have distributed the CA setup package you generated in Step 2 to the users on your Grid, they will need to request a certificate. This is done using the command

```
# grid-cert-request
```

After you have entered a passphrase for the private key, the command will provide you with the following information:

- *Your Certificate Subject*: As an example, if my account username were C<sam> and my machines were in C<sams-domain.org> domain, the subject for my user certificate request would be

```
/O=Grid/O=SamsLittleCA/
OU=sams-domain.org/CN=sam
```

- *CA Email*: As you will notice, this is the same email address as you entered during CA setup back in Step 2.

This will generate a request for the certificate, which must be transferred to you (the CA) for signing. Once you receive this certificate request, you should do the following:

1 Verify the authenticity of the requestor. If the certificate request file contains a subject that doesn't agree with the sender of the request (by email or other means), the request should be rejected. If you as the CA verify the subject matches the sender, the certificate can be signed. More stringent methods of request verification exist, such as person-to-person or contact by phone.

2 Sign the cerificate request. To do this, you must run the `grid-ca-sign` command, with the certificate request as input. The output of the certificate request is the certificate that must be returned to the user. In our current example, the request generated above is signed as follows:

```
# grid-ca-sign -in user-
cert_request.pem -out
usercert.pem
```

The request and signed certificate by default should be located in the `~/.globus/` directory of the user.

We note that in the future, the Globus Simple CA will be included in the Globus Toolkit distribution. This will allow users to bypass the download and installation steps of that software component.

What if I need a host certificate?

Host certificates are generated the same way as user certificates are except that the C<-host> option is used. On a machine with the hostname C<myhost>, the resulting command would be

```
# grid-cert-request
-host myhost.sams-
domain.org
```

Creating a Proxy Certificate

Once you have acquired a certificate from a CA, you are ready to begin using Grid security tools. Only one step remains: generating a proxy certificate. The proxy can be generated simply by running the command

```
# grid-proxy-init
```

Because the proxy is generated from the user certificate and private key, this command will require the password of the user's private key. Once run, the command writes the proxy certificate to `/tmp/x509up_u[user-id]`, where the `[user-id]` is replaced with your user id for the system. The `grid-proxy-init` command does not automatically verify the validity of the user certificate and will continue to generate the proxy without error even for an invalid (e.g., expired) user certificate. The user can elect to use the `-verify` option to `grid-proxy-init` to verify the validity of the generated proxy certificate. The above command becomes

```
# grid-proxy-init -verify
```

Using the `-verify` option is a good way to find problems with your proxy certificate that might not otherwise appear. Not only will the `-verify` option catch expired or revoked user certificates, but it will also check a user's security config-

Resources

Globus Toolkit

- www-unix.globus.org/toolkit/
- www-unix.globus.org/toolkit/documentation.html

Globus Simple CA

- www.globus.org/security/simple-ca.html

uration, which includes verifying that the CA's certificate is installed on the machine. These are all checks that secured Grid applications will perform anyway, so using the `-verify` option can be seen as a trial run for the security components of the Globus Toolkit.

Where to Go from Here

We've just outlined basic credential management of Grid security with the Globus Toolkit. There are many directions to go from here. While the Globus Simple CA does provide useful functionality for many simple Grid environments, it may not provide all the functionality you require. Drawbacks of using the Simple CA include the following:

1 The CA you create and manage from the Globus Simple CA package is trusted only by a small group of users (namely, you, and anyone you convince should trust it as well).

2 The Globus Simple CA is designed with simplicity in mind. There is no central user interface for certificate management, and (although possible to extend) the Globus Simple CA does not include specific tools for certificate revokation.

If you see the potential of constructing your own Grid with a group of organizations, you will want to seriously consider the various CA software choices. Commercial and the more serious noncommercial CAs will generally publish a certification practice statement detailing the certificate issuance process. This document may help in your evaluation. Also, if you are aware of any legal issues, you may want to enlist the help of a lawyer. Numerous different CAs exist, and picking a CA that meets your needs can be a tricky business that should be undertaken with care.

Globus Toolkit is a registered trademark held by the University of Chicago.

This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38; by the National Science Foundation; by the NASA Information Power Grid program; and by IBM.

Sam Meder coordinates development of the Globus Toolkit security and Grid services core infrastructure areas. He can be reached at meder@mcs.anl.gov.

Sam Lang implemented and continues to provide support for the Globus Simple CA software. He is currently an active developer on Globus Toolkit. He can be reached at slang@anl.gov.