UvA UNIVERSITEIT VAN AMSTERDAM

System and Network Engineering

RP1

Troubleshooting Grid authentication from the client side

Adriaan van der Zee

2009-02-05

NIKHEF

Abstract

This report, the result of a four-week research project, discusses the operation and interactions of basic Grid components that are involved in executing jobs on the Grid. It identifies problem conditions and provides a script that tries to identify some of the failure conditions on the User Interface machine to the Grid.

Users create proxy certificates from their personal X.509 Grid certificates to interact with the Grid. A VOMS server should be contacted to obtain VOMS extensions for the certificate, which is necessary to submit a job to a WMS. The WMS needs a delegation proxy certificate to be able to forward the job to a CE. These are some of the interactions that are explained in more detail.

General failure conditions for Grid authentication are discussed, as well as problems specific to UIs. Time skew between systems, expired certificates and CRLs and incorrect or expired proxy certificates are examples of failure conditions that can occur.

A command line tool for the UI, grid-auth-verify.sh, has been developed as part of this research project. This tool performs a number of checks on the UI to identify some of the discussed failure conditions.

The success of the tool, and the possible need to extend it with checks related to communication with foreign hosts need to be determined by evaluation of usage by real Grid users.

Acknowledgements

From NIKHEF I would like to thank Jan Just Keijser and David Groep for their guidance and supervision, and Dennis van Dok and Oscar Koero for their technical and operational input and help.

From the UvA I would like to thank Cees de Laat for his non-intrusive coordination of this project.

Table of contents

Abstract 2
Acknowledgements 3
Table of contents 4
Glossary 5
1 Introduction 6
1.1 Background information about the subject 6
1.2 General description of the project 6
1.3 Problem definition and research questions 7
1.4 Outline of this report 7
2 The Grid 8
2.1 X.509 and proxy certificates 9
2.1.1 The VOMS server 9
2.1.2 The MyProxy server 10
2.2 The User Interface 10
2.3 The Workload Management System 11
2.4 Computing Elements 11
2.5 Logging and Bookkeeping service 11
2.6 Storage Element 11
3 Authentication failure conditions 12
3.1 Unknown CA 12
3.2 Proxy or host certificate expired 12
3.3 Proxy certificate not yet valid 13
3.4 CRL out of date 13
3.5 VOMS attributes missing 13
3.6 UI misconfigurations 13
4 The tool: grid-auth-verify.sh 14
4.1 Date check 14
4.2 UI environment check 14
4.3 Trusted CA certificates directory check 15
4.3 User certificate verification 15
4.4 Proxy certificate chain verification 15
4.5 Proxy certificate content check 15
4.6 Grounds not covered by the tool 16
Conclusions 17
Future work 18
Bibliography 19
Appendices 20
Appendix I – Example output of the grid-auth-verify.sh script 20
Everything OK 20
Proxy almost expired 20
Proxy expired 21
No trusted CA certificate directory found 21

Glossary

CA	Certification Authority
CE	Computing Element, manages clusters of WNs
CRL	Certificate Revocation List
Grid	The grid infrastructure NIKHEF is part of[4]
GSI	Grid Security Infrastructure
IS	Information System, keeps track of the utilisation/availability of resources
Job	Computer program to be executed on the Grid
LB	Logging and Bookkeeping, keeps track of the status of jobs capabilities in VOs
MyProxy	Server that securely stores medium-lived proxy certificates that can be used to renew short-lived proxies on trusted hosts
SE	Storage Element, manages a storage cluster
UI	User Interface, machine capable of submitting jobs to the Grid
VO	Virtual Organisation
VOMS	Virtual Organisation Membership Service, grants users membership, roles and
WMS	Workload Management System, accepts jobs from the UI
WN	Worker Node, executes jobs

1 Introduction

This report is the result of the first of two four-week research projects that are part of the one-year curriculum of the master study System and Network Engineering from the University of Amsterdam (UvA). This research project has taken place at NIKHEF Institute for Subatomic Physics, in the department that maintains and further develops NIKHEF's part of the Worldwide LHC Grid infrastructure. The goal of this research was to be able to develop a tool that can be used to troubleshoot authentication failures that take place in the Grid.

1.1 Background information about the subject

NIKHEF maintains a tier-1 site of the Worldwide LHC Computing Grid[6], which is meant to process and store portions of the data that the LHC[7] is planned to produce. However, this computer grid extends beyond the LHC experiments to other fields of science, such as bio-informatics and medical sciences. The increased number and diversity of users requires improved user-friendliness and troubleshooting capabilities of the grid middleware.

One of the areas where users are faced with problems is authentication. An implementation of the Grid Security Infrastructure (GSI)[8] takes care of authentication based on a Public Key Infrastructure (PKI) with X.509 certificates. Before a computing job from a user can run on the Grid, many intermediate systems get involved, needing to authenticate the user and/or each other, based on (a delegation of) the users certificate. Practice has shown that authentication can fail on a number of different levels, for a number of different reasons, which remain unclear to the user, who only sees that its job has failed.

Authentication between various systems in the Grid is needed to gain access to resources. The User's identity, which is contained in a personal X.509 certificate, is used to determine which Grid resources a user is allowed to access, and what priority a user has compared to other users. X.509 proxy certificates[9] are used by the user for single sign-on purposes, as well as to create delegations for systems in the Grid. Such a delegation proxy certificate enables a system in the Grid to gain access to resources on the user's behalf, and can even be delegated deeper into the Grid.

Because the distributed nature of the Grid, which spans multiple physical and administrative domains, it is challenging to identify an authentication failure. This is partly due to the fact that the Grid middleware is not homogenous either, as different applications have been written by different developers from different organisations.

1.2 General description of the project

The current situation where Grid users are faced with authentication failures which cannot be identified easily and can therefore also not be resolved easily, need to be improved. Due to the complexity and heterogeneity of the Grid middleware it is not to be expected that clear failure reports will be propagated from the source of the problem to the user properly.

As authentication, even between remote systems, with GSI is always based on a (delegation of) a users certificate, it might be possible for a user to contact different components of the Grid directly to test authentication. For this to work it is necessary to determine which Grid components use GSI authentication during job submission and execution on the Grid. Furthermore, it should also be possible to identify the actual physical systems that are being used for a particular job that fails.

Ideally, all tests a user can do to identify authentication problems should be integrated into a single executable or script that a user can run whenever it faces a job failure. The information that this script produces should help resolving the problem, either at the user side, or by pin-pointing the probable cause and/or location where the problem occurred.

1.3 Problem definition and research questions

To what extent can authentication failures in the Grid be identified and resolved from the client side?

A number of sub-problems are:

- What are the possible causes of GSI authentication failures?
- Which grid components are involved in GSI authentication for a standard job submission and execution?
- How can a client determine which systems are probable causes of authentication failure for a job?
- Is it possible for a client to test authentication by contacting such systems directly?

1.4 Outline of this report

In section 2 a general overview of the Grid and its use is given. Proxy certificates and their role in Grid authentication are discussed in subsection 2.1, and subsection 2.2 to 2.6 are about the main Grid components and their interactions.

In section 3 a number of possible failure conditions are discussed, as well as possible ways to prevent or detect them.

Section 4 is about the tool that has been developed for the UI, to detect a number of possible failure conditions as described in the previous section 3.

2 The Grid

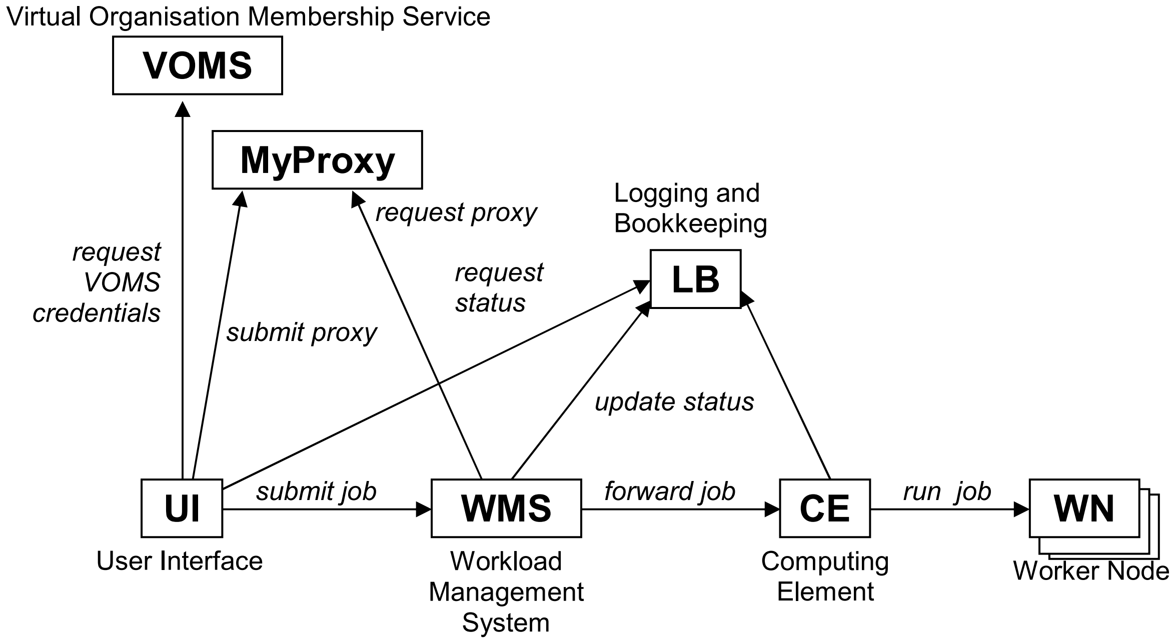

Figure 1 - Simplified schematic of the Grid from the user's perspective

The Grid infrastructure NIKHEF is involved in consists of thousands of CPUs, hard disks and tapes intended for performing complex computational tasks, and storing vast amounts of data. By splitting up tasks that would take months or even years to complete on a single PC, and running them in parallel on the Grid, they can be completed in a fraction of the time. Each of such a part of a task is called a job in Grid terminology.

Jobs are submitted by end users to a Workload Management System (WMS), the entry point to the Grid. In order to submit jobs to the WMS a user needs to have specific software installed on its computer that is part of the Grid middleware. A system with all these necessary tools installed is referred to as a User Interface (UI) to the Grid. An actual machine that executes jobs is called a Worker Nodes (WN). These WNs are organised in clusters, and one or several clusters are managed by a Computing Element (CE).

CEs are designated to schedule jobs belonging to specific users or groups of users to the WN clusters that it manages. Users are organised in Virtual Organisations (VOs) based on their real organisation, area of research, specific experiment, etc. Within a VO a user can have special roles and capabilities to make a finer grained distinction between users. CEs report their status to the Information System (IS), and based on this information WMSs can distribute incoming jobs to the available CEs.

Storage Elements (SEs) are systems that manage storage space intended to store large amounts of data, which might be the input or output of jobs.

Once a job has been submitted from a UI to a WMS its status is continuously being updated to the Logging and Bookkeeping (LB) service which the user can query to monitor the progress of its jobs.

Within the Grid the identity of the user that submitted a job plays a crucial role, as this determines the capabilities and priority of jobs, and is used for accounting as well. Not only does a user need to prove its identity when submitting a job to a WMS, proof of identity needs to accompany the job further into the Grid to CEs as well. In order to achieve this X.509 certificates[2] are issued to users, which use it to create a proxy certificate[9] for single sign-on, and delegation further into the Grid.

Two more Grid services are provided for identity management: the Virtual Organisation Membership Service (VOMS), a service that assigns VO membership, roles and capabilities to users, and MyProxy, which a WMS can use to renew proxy certificates.

In subsection 2.1 identity management with X.509 and proxy certificates is explained as well as the role and interactions of the VOMS and MyProxy servers, and in section 2.2 to 2.6 UI, WMS, CE, LB and SE are discussed.

2.1 X.509 and proxy certificates

Anyone who wants to access the Grid needs to have a personal X.509 certificate issued by a CA that is trusted by the Grid. Such a X.509 certificate consists of a public certificate and a private key[2]. This key should be accessible only by the owner, and be protected with a passphrase. The key can be used by the user to prove ownership of the certificate, and thus prove its identity. A Grid certificate is typically valid for a year, and if the private key were to be compromised, the certificate should be revoked, which renders it invalid permanently.

The personal certificate, however, is not used by the user to authenticate on the Grid directly, but instead proxy certificates[9] are being used. A proxy certificate is a compromise between security and user-friendliness. The proxy's key is not protected with a passphrase, but to limit the possibility of abuse the proxy is only valid for a limited amount of time, typically twelve hours. A proxy certificate is created by generating a new certificate, and signing it with the key of the original X.509 certificate. The newly created proxy certificate can be used to prove identity on the grid in the same way the original certificate could have been used.

Proxy certificates are being used to serve two particular purposes: single sign-on and delegation. If a user would use its own certificate to authenticate for each interaction with the Grid, being it submitting a job, or querying the LB for a job's status, it needed to enter its passphrase each and every time. Single sign-on is meant to increase user-friendliness allowing a user to create a proxy certificate, entering the passphrase once, and use the proxy (that is not protected with a passphrase) for the rest of the day.

The other goal of a proxy certificate is delegation. A job is typically submitted to a WMS, who propagates it further into the Grid to a CE. In order for the job to retain its identity the WMS uses a proxy of the user certificate to authenticate with the CE. Such a proxy is generated by the WMS, and signed by the user's proxy certificate.

2.1.1 The VOMS server

In the past a user's identity was enough to be granted or denied access to certain parts of the Grid. Each system in the Grid maintained a gridmap-file that mapped users to local system accounts. This, however, did not scale well for the increasing number of users and organisations, and therefore an extra abstraction layer has been introduced to categorise users from physical organisation into VOs, and in addition to that give them specific VO independent roles and capabilities[10]. These memberships, roles and capabilities are maintained by a VOMS server that is to be trusted by everyone on the Grid.

Users can prove their membership, roles and capabilities to other systems on the Grid by means of attribute certificates, also called VOMS extensions. These VOMS extensions are included in the proxy certificate by the user, and signed by a VOMS server. The user obtains such a VOMS extension by contacting the VOMS server, proving its identity with a proxy certificate, and the VOMS server will return, if granted, the user's request for VO credentials which it has signed. An authenticating system relies on the signature of the VOMS server for the authenticity of the credentials included in the proxy certificate.

2.1.2 The MyProxy server

As already mentioned a proxy certificate has a typical life time of only twelve hours, and any further delegation cannot have a lifetime that exceeds that of the parent. This poses a problem to jobs that require to run longer, because the delegated proxy will have expired before the job has finished. To tackle this problem the MyProxy server has been introduced.

The MyProxy server is intended to store medium length proxy certificates, typically valid for one week. A user can store a proxy certificate on the MyProxy server, and remove it at will. The MyProxy server is supposed to be a relatively secure place to store such medium-term proxy certificates, and is accessible by WMSs. A WMS can use the MyProxy server to renew a user's proxy. The MyProxy server is only supposed to trust requests from machines (WMSs) that have been explicitly configured on the MyProxy server, as to limit the possibility of a compromised proxy being renewed.

2.2 The User Interface

The User Interface (UI) is the machine a user submits its jobs from. This is typically a Linux machine with Grid middleware binaries installed and included into the PATH variable of the user's environment. On the UI the user's X.509 certificate is stored, as well as the proxy certificate the user works with.

From the UI the user contacts the VOMS server in order to obtain VOMS extensions which it can include in its proxy.

Another interaction from the UI is with the MyProxy server in order to store a medium-term proxy. This is optional, but necessary when the user wants to submit jobs that will run longer than the lifetime of the short-term proxy certificate.

The most frequent interactions from the UI are with the WMS and LB service. The WMS is contacted to submit jobs, and the LB service is queried to retrieve status information about jobs.

2.3 The Workload Management System

The WMS is the entry-point into the Grid from the user's point of view. It accepts jobs from users, and sends them through to CEs that advertise availability to certain VOs, based on information from the Information System. Thus, the WMS accepts connections from UIs, and authenticates them with the user's proxy certificate. Furthermore, the WMS needs to get a delegation of the user's proxy in order to be able to forward the job to a CE. For this delegation the WMS creates a new certificate and key pair, includes the original public part of the user proxy certificate, and have it signed by the original user proxy certificate.

Once an eligible CE has been identified by the WMS to forward the job to, the WMS contacts that CE authenticating itself with its delegated proxy from the user. If the CE accepts the job it will take care of job execution from that point on.

The WMS contacts the LB to log status messages about the job.

2.4 Computing Elements

The CE accepts jobs from a WMS based on the user credentials the WMS sends. Once it has accepted a job, the CE will take care of its execution, and it will contact the LB to log relevant status events.

2.5 Logging and Bookkeeping service

The LB stores status information about jobs that it receives from WMSs and CEs. The LB can be contacted by a UI that requests logging information.

2.6 Storage Element

SEs manage storage clusters, intended to keep large amounts of data that come as input or output of jobs. SEs are contacted by WNs that need to access data, as well as by UIs that manage the data. Interactions with SEs however, are very job-specific, and are therefore beyond the scope of this research. Furthermore varying implementations of interactions and authentication with SEs exist in different implementations.

3 Authentication failure conditions

As described in section 2 the Grid consists of several components that need to authenticate each other with mutual authentication. In many cases one side of the authentication is with a user's proxy certificate, and the other side is with an X.509 host certificate, issued by a trusted CA. Several conditions can occur that result in an authentication failure. The following subsections discuss a number of such failure conditions that are common in the Grid.

3.1 Unknown CA

The basis of all X.509 based authentication is an anchor of trust that should be installed on each authenticating host. The authenticated system presents an X.509 certificate, or a proxy certificate chain, but the original X.509 certificate should be signed by a CA that is trusted by the authenticating system. In practice this means that all systems participating in the Grid have a designated directory containing all certificates of the CAs that should be trusted. On a UI this directory is usually stored in the environment variable X509_CERT_DIR.

If this directory is missing, or not containing all certificates of trusted CAs authentication will fail.

3.2 Proxy or host certificate expired

As already mentioned in subsection 2.1 a proxy certificate has a very limited lifetime, and is therefore prone to expiration. If a user tries to use an expired proxy to access the Grid, access will be denied clearly, but less visibly, a WMS that tries to contact another system on the user's behalf could be facing the same problem, which is less obvious to the user. One way to overcome the latter problem is to make use of the MyProxy service as described in 2.1.2.

Apart from the user's proxy that can expire, any host certificate from a system in the Grid can expire if the administrator fails to renew the certificate in time. This will occur less frequently because host certificates are typically valid for at least a year, but if such a situation should occur, it is harder to detect because it is less common.

Another possibility is that there is disagreement about whether or not a certificate has expired. This is possible when the clocks of the authenticating systems are out of sync. Because an authenticating system checks the validity of a certificate relative to its own system time, it is possible that one deems a certificate expired, while the other does not. This situation can also result in the reverse effect, as explained in the following subsection 3.3 about certificates that are deemed not yet valid.

3.3 Proxy certificate not yet valid

Because proxy certificates are commonly used just after they are created, clock skew between authenticating systems might result in a proxy being rejected because the certificate is not yet deemed valid by the authenticating host. Because a proxy certificate's validity is expressed as start and end timestamps based on the clock of the system that creates the proxy certificate, an authenticating system might find it is not yet valid if its clock is behind that of the originator of the proxy. To allow for some clock skew a proxy certificate is by default created with a start time five minutes in the past, but if the UI would be more than five minutes ahead, a failure condition can be triggered[1].

Therefore it is important that all systems in the Grid should be more or less in sync with each other, something that can be achieved by synchronising all systems to network time through NTP[5].

3.4 CRL out of date

X.509 certificates are issued by trusted CAs, whose signature warrants the certificate. However, there is also a mechanism to revoke certificates which are reported to be compromised. This revocation works through so called Certificate Revocation Lists (CRLs). A CA publishes a CRL which should be installed next to the CA's certificate on all systems performing authentication on the Grid. In order to force these CRLs to be updated regularly they have an expiration date, after which they are not valid anymore. If a system would fail to renew a certain CRL in time, it will expire. A system should not accept any certificate signed by a CA who's CRL has expired, because it cannot determine if the certificate was revoked or not.

3.5 VOMS attributes missing

In order for a UI to contact a WMS it needs to present proof of VO membership, roles and capabilities in the form of VOMS extensions as described in 2.1.1. If the user's proxy certificate does not contain at least one such a VOMS extension the WMS will deny access.

3.6 UI misconfigurations

Any system in the Grid that is misconfigured can cause authentication to fail. However, because the UI is the most unstable and unpredictable environment, misconfigurations are most likely to occur there.

First of all, the user's X.509 certificate needs to be properly installed on the UI, and the system time should be reasonably in sync. Furthermore, the trusted CA certificates along with their CRLs should be installed, and accessible. The command line tools for Grid interactions should be available in the PATH environment variable, and these tools need to be used properly in order to create a valid proxy with included VOMS extensions.

4 The tool: grid-auth-verify.sh

The goal of this research project was to be able to develop a tool that can be run on the UI that tries to detect as many as possible failure conditions as described in section 3. Some of these checks, as far as the time constraint of this project allowed, have already been implemented in a shell script grid-auth-verify.sh. The script can take command line parameters to overrule default or environment variable for the trusted CA certificates directory, the user certificate location, and the proxy certificate location. The script will output info, warning and error messages, of which the latter will cause the script to abort. All these messages are also stored in a log file.

The script makes use of standard Linux tools and specific Grid middleware tools, all of which are expected to be installed on the UI already. In addition the script makes use of a custom C-program grid-proxy-verify[11] that is to be provided along with the script.

The following subsections explain checks that this script performs, and the last subsection discusses some of the features that have not yet been implemented, as well as challenges that prevent easy and or generic implementation.

Please refer to Appendix I for example output of the script.

4.1 Date check

As described in subsections 3.2 and 3.3 time plays an important role in the process of (in)validating a (proxy) certificate. Because it is hard, if not impossible, to check the local system time of an arbitrary system in the Grid, the best thing the tool can do is to make sure the UI itself is synchronised to a reliable NTP[5] source.

In order to verify that the time of the UI is within reasonable sync the `ntpdate` command, which is common on Linux machines acting as UI, is used to measure the time offset to an NTP server. The tool uses one of Surfnet's NTP servers that are accessible to the world[3]. If the offset exceeds the predetermined value (stored in a constant in the script) a warning is given to the user that the UI is out of sync.

4.2 UI environment check

A very basic check is performed to verify that the Grid middleware command line tools are available in the PATH environment variable. If this is not the case, the script will exit with an error, as the Grid middleware tools need to be available for any successful Grid interaction.

4.3 Trusted CA certificates directory check

As explained in subsection 3.1 all systems need to have the certificates of all trusted CAs installed. On the UI this check consists of verifying that the directory that is supposed to contain these certificates exists, and contains at least one certificate. The directory that will be checked will be the one given as command line parameter, the one stored in the X509_CERT_DIR environment variable, or the default location /etc/grid-security/certificates/ respectively. If no valid directory is found the script will exit with an error, as no Grid authentication can take place without pre-installed and accessible CA certificates.

4.3 User certificate verification

To verify the integrity and validity of the user certificate itself the verify option of openssl is used on the original X.509 user certificate. The location of this certificate can be given as a command line parameter, or else the default location in the `.globus` directory under the user's home directory will be used. If the certificate is not found, or if it did not verify correctly, a warning will be given, as the user certificate, however crucial for generating a proxy certificate, is in itself not involved in grid authentication.

4.4 Proxy certificate chain verification

In order to check the complete certificate chain of a given proxy, as well as the integrity of the proxy certificate a custom C-program `grid-proxy-verify`[11] is used. Because this program is not installed on UIs by default, it is to be provided along with the `grid-auth-verify.sh` script. The script will first check if a compiled version of the `grid-proxy-verify` program is present, and if not, it will try to locate the source, and compile it. If any of these steps fails a warning will be given to the user, and the grid-proxy-verify program will be skipped.

The result of the grid-proxy-verify program, along with any additional output errors is given back to the user. If problems have been found with the certificate chain or integrity of the proxy certificate, the script will exit with an error, because the proxy clearly is not suitable for use on the Grid.

4.5 Proxy certificate content check

Typical things that can be amiss with a proxy certificate are expiration (see subsection 3.2) and the absence of VOMS extensions. The `grid-auth-verify.sh` script uses the UI command line program `voms-proxy-info` as its source for information about the contents of the certificate.

From the output of `voms-proxy-info -all` information about the validity of the certificate, and included VOMS extensions is read. If any of those expire within an hour a warning is returned to the user. The same output is also used to check if any VOMS extensions are present, and if not, another warning is given to the user.

4.6 Grounds not covered by the tool

Unfortunately the `grid-auth-veryfy.sh` script does not contact any other machines in the Grid, which somewhat limits its effectiveness to troubleshoot problems that occur deeper into the Grid. This is partly due to the limited time allocated for this project, but also because of differences in connection and authentication implementations between the different Grid components. Some serious challenges will have to be overcome to implement checks beyond the UI itself, as many of the grid components interact differently with each other.

One obstacle is the fact that the openssl program, which could be particularly useful for troubleshooting authenticated Grid connections, is not yet capable of dealing with proxy certificates. The version installed on UIs by default is 0.9.7d, whereas proxy support is included as of version 0.9.7g[12].

Another problem is that the Grid systems involved are very job-specific, and rather unpredictable. Therefore input from the LB service will have to be interpreted to find out which systems should be troubleshooted, something that is not trivial.

Furthermore, different versions of different components might use different connection and authentication strategies. The currently common LCG CE[13] still uses an old Globus GSI implementation[8] is being replaced by the CREAM CE[14] which uses a more modern Web Services based interface.

One obvious check that has not been done by the script on the UI is to check CRLs. Investigation of the CRLs installed on several UIs has shown that some are out of date, but this does not necessarily lead to authentication failures, as not all CRLs will be used by the UI. Which CRL should be used depends on the CA that signed the certificate of the authenticated machine, and thus cannot be determined beforehand. As different Grid components use different authentication connection implementations, it seems not to be possible to request the certificate from all different Grid components in a generic way, and therefore a CRL check could not be included in the script.

All these factors make it very challenging to make a generic tool to verify Grid authentication beyond the UI.

Conclusions

A multitude of Grid components is involved with a single job. The UI needs to contact the VOMS server to get VOMS extensions it needs to be able to contact the WMS. The WMS needs to be delegated a proxy certificate from the user in order to forward it to a CE for execution on a WN. UI, WMS and CE communicate status information with an LB, and SEs could be involved as well.

In each of these interactions authentication failures can occur, for different reasons. However, problems are most likely to be caused by problems at the UI, as this component is not necessarily maintained by a Grid system administrator. On the UI things can be wrong with the user's X.509 certificate, its proxy and VOMS extensions, the system clock can be adrift, and problems in the environment, like the trusted CA certificates directory or CRLs can be amiss.

The tool that has been developed for this research project, grid-auth-verify.sh, tries to identify a number of faults at the UI side. These checks include a date synchronisation check, environment checks, verification of the certificate and proxy certificate chain, as well as the content of the proxy certificate like the presence and validity of VOMS extensions.

The tool aims to provide meaningful messages to the user about succeeded or failed tests, which should help a user who experiences problems with interacting with the Grid.

Future work

The effectiveness of the grid-auth-verify.sh script that is the result of this research project should be evaluated by having users use it in real problem scenarios. After the effectiveness of the script as it is now is established decisions can be made to extend the tool to do further checks on the UI itself, like verifying CRLs, and/or extensions that check connections to one or more external Grid services.

Challenges that have to be overcome to develop the tool further are manifold, for one thing a way should be found to generically interpret output from LB the service to determine which systems should be contacted. Contacting different systems is hindered by the fact that different Grid components and even different versions of the same Grid components (see subsection 4.6 for different CEs) implement the authentication process differently. And if connections are to be troubleshooted with openssl, the feasibility of including an openssl version capable of dealing with proxy certificates[12] should be investigated.

Any further extensions to the script should depend on the success of the current script, and the necessity to be able to troubleshoot beyond the UI.

Bibliography

[1] Authentication failures due to clock skew,
http://www-unix.globus.org/mail_archive/discuss/2002/01/msg00055.html

[2] RFC3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,
http://www.faqs.org/rfcs/rfc3280.html

[3] Surfnet's NTP service,
http://www.surfnet.nl/en/diensten/netwerkinfrastructuur/Pages/ntp.aspx

[4] NIKHEF's Grid website, http://www.nikhef.nl/grid/

[5] RFC1305 - Network Time Protocol (Version 3) Specification, Implementation and Analysis http://www.faqs.org/rfcs/rfc1305.html

[6] LHC Computing Grid, http://lcg.web.cern.ch/lcg/

[7] Large Hadron Collider, http://lhc.web.cern.ch/lhc/

[8] Globus GSI implementation, http://www.globus.org/security/overview.html

[9] RFC3820 - Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile
http://www.faqs.org/rfcs/rfc3820.html

[10] R. Alfieri et al., From gridmap-file to VOMS: managing authorization in a Grid environment, Future generation Computer Systems 21 (2005) 549-558, Elsevier 2005

[11] Jan Just Keijser, 'grid-proxy-verify' utility, http://www.nikhef.nl/~janjust/proxy-verify/

[12] Proxy certificate support in openssl,
http://www.mail-archive.com/openssl-announce@openssl.org/msg00057.html

[13] LCG CE, http://glite.web.cern.ch/glite/packages/R3.1/deployment/lcg-CE/lcg-CE.asp

[14] CREAM CE, http://grid.pd.infn.it/cream/

Appendices

Appendix I – Example output of the grid-auth-verify.sh script

The following examples have been performed on icarus, a Linux workstation within NIKHEF that has been prepared to act as UI.

Everything OK

```
bash-3.00$ ./grid-auth-verify.sh
INFO: Trying to check time difference with chime2.surfnet.nl
INFO: Local time differs 0 seconds from network time, which is within set
limit of 60
INFO: Trying to locate directory with trusted certificates
INFO: Will use /global/ices/lcg/glite3.1.23/external/etc/grid-
security/certificates from evironment variable X509_CERT_DIR for trusted
certificates
INFO: Trying to verify user certificate
INFO: Will use /user/adriaanz/.globus/usercert.pem as user certificate
INFO: User certificate verification succeeded
INFO: Trying to verify proxy certificate chain
INFO: Will use /tmp/x509up_u7899 as proxy certificate
INFO: Proxy certificate chain verified succesfully
INFO: Trying to check proxy content
INFO: No irregularities found in proxy contents
```

Proxy almost expired

```
bash-3.00$ ./grid-auth-verify.sh
INFO: Trying to check time difference with chime2.surfnet.nl
INFO: Local time differs 0 seconds from network time, which is within set
limit of 60
INFO: Trying to locate directory with trusted certificates
INFO: Will use /global/ices/lcg/glite3.1.23/external/etc/grid-
security/certificates from evironment variable X509_CERT_DIR for trusted
certificates
INFO: Trying to verify user certificate
INFO: Will use /user/adriaanz/.globus/usercert.pem as user certificate
INFO: User certificate verification succeeded
INFO: Trying to verify proxy certificate chain
INFO: Will use /tmp/x509up_u7899 as proxy certificate
INFO: Proxy certificate chain verified succesfully
INFO: Trying to check proxy content
WARNING: Your proxy or one of its attributes is valid for less than one
hour
```

Proxy expired

```
bash-3.00$ ./grid-auth-verify.sh
INFO: Trying to check time difference with chime2.surfnet.nl
INFO: Local time differs 0 seconds from network time, which is within set
limit of 60
INFO: Trying to locate directory with trusted certificates
INFO: Will use /global/ices/lcg/glite3.1.23/external/etc/grid-
security/certificates from evironment variable X509_CERT_DIR for trusted
certificates
INFO: Trying to verify user certificate
INFO: Will use /user/adriaanz/.globus/usercert.pem as user certificate
INFO: User certificate verification succeeded
INFO: Trying to verify proxy certificate chain
INFO: Will use /tmp/x509up_u7899 as proxy certificate
ERROR:  Verifying proxy: Proxy certificate expired.
ERROR:  Verifying certificate chain: certificate has expired
```

No trusted CA certificate directory found

```
bash-3.00$ ./grid-auth-verify.sh
INFO: Trying to check time difference with chime2.surfnet.nl
INFO: Local time differs 0 seconds from network time, which is within set
limit of 60
INFO: Trying to locate directory with trusted certificates
ERROR: Cannot find trsted certificates directory in either the environment
variable X509_CERT_DIR, or /etc/grid-security/certificates or
/user/adriaanz/.globus/certificates
```