Usability Issues in WLCG Security

WLCG Technology Evolution Group on Security

Editor: Von Welch (vwelch@indiana.edu)

April 30, 2012

Version 0.3

1 Introduction

Usability is a key factor in the security of a system, as a lack of security usability will lead to improperly deployment, misuse or avoidance of security mechanisms. In this document we describe a number of usability issues identified by the WLCG Technology Evolution Group (TEG) on Security and make some recommendations on addressing those issues. We divide the list of issues into problems typically seen by scientific researchers who comprise the WLCG's users, and the WLCG administrators.

1.1 From the Researcher Perspective

- **Credential management**: Some WLCG users have trouble with credential management throughout the lifecycle, including obtaining a credential (including the time required) and maintaining it securely without losing it or forgetting the encryption pass phrase. This problem is increased when a user attempts to undertake multiple activities, which require different VOMs[1] attributes; This requires juggling multiple proxies, with different VOMS attributes in each, and brings the risk of getting them mixed up, accidentally overwriting them, using them when they are very close to expiring, etc.

- **Incoherent proxy storage on complex systems**: Proxy credentials are by default stored in */tmp*, which for some systems (e.g. clusters) is not shared across the whole system, leading to incoherencies in availability of the proxy to activities.

- **Supporting both web and non-web authentication**: In order to use X.509 credentials for both web browsers and non-web applications, the credentials must be exported from or imported into a web browser. This can be tricky to get the details right and onerous for credentials with short-life time (e.g., as issued from a KCA or MyProxy online CA). Additionally, support for RFC 3820 proxy certificates by web applications is weak.

[1] https://twiki.cern.ch/twiki/bin/view/LCG/VOMS

- **Lack of internationalization**: Support for internationalization of names in X.509 distinguished names is not complete and use of non-ASCII characters can cause failures.

1.2 From the Administrator's Perspective

- **Managing revocation**: Revocation of certificates is accomplished through certificate revocation lists (CRLs), which have finite lifetimes (typically in the range of a week to a month). When a CRL expires, software still using that CRL will fail safe and fail all authentication attempts. This means that timely updates of CRLs are an operational necessity and any failure of a certificate authority to produce a new CRL or a relying party to obtain and properly install it results in a service outage.

- **Expired host and service certificates**: Certificates used by hosts and services have finite lifetimes, typically a year or small number of years. After they expire, clients attempting to use those hosts and services will experience failures. Currently administrators are responsible for "manually" monitoring these certificates and renewing them before expiration to provide continuous service. The case of a cluster with tens or hundreds of certificates may pose issues of scale.

- **Managing authorization policies**: Authorization is controlled by policies that are encoded in a number of different places: grid-mapfiles, CA signing policies, VOMS, GUMS, etc. Authoring and maintaining all of these to coherently represent the mission of the projects and VOs is a complicated, error-prone task.

- **Client authorization of hosts and services**: *placeholder, to be completed* Clients expect services to identify themselves with host certs by default and therefore may need special configuration to accept anything else, viz. a service certificate.

- **Inconsistent user banning mechanisms**: It is occasionally necessary to "ban" a user, or suspend their ability to access resources, due to a suspected or confirmed security incident, misbehavior of the user or their software, or other administrative reasons. This differs from certificate revocation, which is meant to globally revoke a user's authentication credentials, and may not be an appropriate vehicle due to the scope or longer-term authentication effects. Similar to more general problems of managing authorization policies, software services differ in how they allow for banning or whether they even do, leading to a inconsistent, error-prone process.

- **Mixing of authentication and authorization**: Proxy certificates, through their use of impersonation (allowing one entity to appear as another entity)

to support delegation, are inherently a mix of two security concepts - authentication (identification of an entity) and authorization (what is that entity allowed to do). This creates some usability problems such as masking of a delegatee's identity in logging (they get logged as the delegator) and confusion between the use of entity names and VOMS attributes in authorization policies.

- **Lacking debugging and forensics**: Debugging of failures is a challenge caused by the multiple software stacks with different error propagation characteristics and capabilities, and the distributed nature of the system (and the resulting logs) and its users community. This causes challenges in accessing needed log and configuration data for debugging (assuming it exists), and interpreting it as logging across different software components is not standardized in terms of content, syntax or semantics.

- **Inconsistent proxy certificate implementations**: Inconsistent implementations of support for proxy certificates lead to incompatibilities and difficulty in other administrative tasks, such as infrastructure upgrades. An example is the lack of RFC 3820 proxy certificate support in some software, leading to difficulties in removing dependencies on weak cryptographic algorithms.

- **X.509 validation overhead**: Currently the validation of X.509 credentials is done for each network connection within the WLCG. Each of these validations is equivalent to the validation done for a TLS/SSL handshake as part of a web HTTPS connection. And while Google has done significant work in minimizing this overhead [1], the aggregate load across the WLCG may be significant.

2 Recommendations

We now turn from problem identification to recommending improvements to address those problems. We sort the recommendations into short-term, defined as something that could be put into place in less than 2 years, and long-term, or something that would require more than 2 years to put into place.

2.1 Short-Term

In general, effort should be made to "hide" PKI/X.509 from the end-users of the system as much as possible. Options here include:

- **Minimizing the enrollment process by leveraging existing sources of identity and authentication**. For example, leveraging existing member

authentication services either directly, e.g., the IGTF MICS profile [2], or through federated identity, e.g. the CILogon service[2].

- **Minimizing the credential management process through the use of short-lived credentials**. In environments where the use of credentials in web browsers is not needed, short-lived credentials (i.e., those with lifetimes of less than 10^6 seconds or roughly 11 days) as defined by the IGTF SLCS profile [3] in combination with the leveraging of existing authentication mechanisms offers users easy to obtain credentials that are in effect "disposable," removing burdens for long-term curation, migration between computers, renewal, etc. (While such credentials would work in a web-based environment, the frequent routine of importing them into a browser would be onerous.)

Other improvements for researchers include:

- **Tools for multiple credentials**. Some users need to juggle multiple credentials with different attributes or from different CAs for connecting to different services with different trust policies, or for acting in different roles. Creating and managing each of these credentials involves a workflow of creating the credential and associated attributes (e.g., grid-proxy-init, myproxy-logon, voms-proxy-init), selecting an appropriate credential for each task (e.g., use credential A for services M,N, and O, and credential B for services X,Y, and Z), monitoring credentials for expiration and renewing them before use if expiration is pending soon, etc. A set of tools or improvements to existing client software to automate the maintanence of multiple "roles" each with a different credential. switch easily between those roles, renew local credentials when they are close to expiring and similar tasks would be beneficial.

From the administrator perspective, short-term improvements include:

- **Tools for service credential lifecycle management.** Develop a set of supporting tools to help monitor and renew host and service credentials to prevent expiration failure and reduce workload during normal operation. Automatic renewal may not be possible in many cases, but a cron job could at least alert the admin on time.

- **Improved revocation.** In addition to short-term tools for improved CRL management (either removing CRLs what are expired or warning administrators), an analysis should be undertaken with regards to the trade-off between availability and security with regards to the behavior of software in "failing safe" when an expired CRL is encountered. Allowing configuration

[2] http://www.cilogon.org/service

of behavior such that software will continue to function with an expired CRL but produce strong warnings may be desirable in some situations and reduce availability failures. Additionally, continued deployment of site-controlled banning (e.g., GUMS/SAZ[3], Argus[4]) would benefit in providing sites the ability to ban users in a more timely manner than CRLs as will as enforce site-specific policy.

- **Standards for logging**. Having a set of defined use cases, derived requirements and ultimately standard mechanisms (e.g., Syslog) as well as syntax and semantics (e.g., [6,7]) for logging would aid in a variety of activities from debugging to cybersecurity incident response.

- **Usability Evaluation**. Information about usability, including the basis for this report, is largely anecdotal and probably skewed towards vocal minorities. It would be beneficial to have some objective evidence, through a professional survey of the user community or have having an evaluation of the usability of grid security and relevant tools performed by expert in the area of usability and human computer interaction.

Prioritization of short-term recommendations:

1. Tools for service credential lifecycle management
2. Improved revocation
3. Standards for logging
4. Leverage existing identity sources to minimize enrollment
5. Usability evaluation
6. Minimize credential management via short-lived credentials
7. Tools for multiple credentials

2.2 Long-Term

Two recommendations in the long-term for improvement are:

2.2.1 Consider a coherent, cross programming language grid security library

While there are some very popular security libraries (e.g. OpenSSL), there is no single library that supports grid security across the range of programming languages used by the grid community. This is in no small part because there are few security libraries in general that support multiple programming languages and ones that do are very limited in functionality, such as keyCzar[5] which has no support for X.509.

[3] http://computing.fnal.gov/docs/products/voprivilege/index.html
[4] http://glite.cern.ch/glite-ARGUS/
[5] http://code.google.com/p/keyczar/

This brings about challenges including steep learning curves for changing programming language, interoperability between languages due to quirks in implementations, and a lack of features (e.g., hash algorithms) in some implementations hindering upgrades. Having a coherent implementation of grid security features (e.g., RFC 3820 Proxy Certificates [4]) in appropriate languages and frameworks would address many incompatibility challenges being faced today and make development of new grid-based applications both easier and more secure.

It is noted however, there are some arguments against this approach that would need to be considered:

- Ideally existing library maintainers would support needed grid functionality. However, experience has shown this is not easy to achieve without significant resource contribution towards that maintenance (providing staff or funding to the maintenance team).
- Having a separate grid security library may cause unexpected behavior when porting applications from other libraries, and effort in dealing with that needs to be considered.

2.2.2 Consider re-implementation leveraging experiences

Over the past fifteen years, there has been much learned about how global-scale grids operate in practice. Rather than incremental improvements described to this point, a next generation may be considered. Aspects to consider would be to re-approach impersonation and alternatives that separate authentication and authorization more cleanly in the delegation process; push public key cryptography to the outer edges and use some sort of session token (e.g., as in the Condor CEDAR implementation [5]) for transaction authentication; implementation of role and group authorization, particularly in the context of the growth of data.

3 References

1. Imperial Violet Blog (authored by Adam Langley, Google Security Engineer). "Overclocking SSL." 25 June 2010. http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html
2. The Americas Grid Policy Management Authority. Member Integrated X.509 PKI Credential Services (MICS), version 1.1. 2 May 2009. http://www.tagpma.org/node/38
3. The Americas Grid Policy Management Authority. Short Lived Credential Services X.509 Public Key Certification Authorities, version 2.1b. 3 February 2009. http://www.tagpma.org/authn_profiles/slcs
4. Steven Tuecke, Von Welch, Doug Engert, Laura Perlman and Mary Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC3820, 2004.

5. Zach Miller, Dan Bradley, Todd Tannenbaum and Igor Sfiligoi. Flexible session management in a distributed environment. 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09). http://dx.doi.org/10.1088/1742-6596/219/4/042017
6. EGEE Security Coordination Group. Middleware Security Audit Logging Guidelines V 0.8. May 14, 2007. https://edms.cern.ch/document/793208
7. Dan Gunter. Logging Best Practices. November 17, 2010. http://netlogger.lbl.gov/home/documents