An online credential management service for InterGrid computing

AUTHOR(S)

Jemal Abawajy

PUBLICATION DATE

01-01-2008

HANDLE

10536/DRO/DU:30018224

2008 IEEE Asia-Pacific Services Computing Conference

An Online Credential Management Service for InterGrid Computing

Jemal H. Abawajy
Deakin University,
School of Information Technology
Geelong, VIC., 3217, Australia.

Abstract

Grid users and their jobs need credentials to access grid resources and services. It is important to minimize the exposure of credentials to adversaries. A practical solution is needed that works with existing software and is easy to deploy, administer, and maintain. Thus, credential management services are the wave of the future for virtual organizations such as Grid computing. This paper describes architecture of a scalable, secure and reliable on-line credential management service called SafeBox for InterGrid computing platform. SafeBox provides InterGrid users with secure mechanism for storing one or multiple credentials and access them based on need at anytime from anywhere.

Keywords: Grid computing, credential management, authentication, authorization, InterGrid Computing, Secuirty.

1. Introduction

InterGrid computing is a computing platform that conglomerates independently managed Grid Services that possibly vary with regard to the type of credentials used to prove user identity (e.g., username/password, X.509, Kerberos, etc.) [1]. However, the security needs of different InterGrid systems vary widely and the authentication mechanisms are similarly diverse. This means, an end-user can potentially have multiple and diverse credentials to access grid resources and services.

Accommodating a diversity of security systems across multiple domains remains a fundamental challenge [2]. Also, the security of a system is often depends on how securely user credentials are managed. Moreover, as grid computing is moving from research into commercial arena, resource owners may require guarantees as to the level of protection user credentials are afforded [8]. Moreover, a credential management service with capability of exchanging different types of credentials has been identified as being critical to the success of the security Grid Services [2, 3, 15].

In InterGrid computing platform, asking the end-users to manage and present credentials manually for each service is tedious, error-prone and potentially insecure [2]. Therefore, allowing users to manage their credentials is risky. Practically, users cannot be expected to obtain and keep track of all the associated certificates and private keys. Most importantly, user managed credentials are error-prone and can lead to significant security and usability problems [3]. In fact, most grid users are mainly interested in seamless and reliable access to grid resources and services without undue regard for the specific security technologies or cryptographic algorithm used for access control mechanisms. Since it is difficult for users to manage their own credentials [3], there is a need for tools that provide for secure storage, management and ubiquitous access to credentials from anywhere at any time.

An online credential repository can provide an alternative to user-managed long-term credentials. Such repositories can provide a well-secured credential storage service that allows credentials to be retrieved easily over the network with appropriate authentication. Several credential management systems with varying capabilities have been designed and developed.

MyProxy [3] uses a central server for credential storage thus scalability and availability is a problem. Moreover, centralized credential management raises significant security concerns as the central server is an attractive target. For instance, it is susceptible to denial of service attack (DoS).

978-0-7695-3473-2/08 $25.00 © 2008 IEEE
DOI 10.1109/APSCC.2008.74

CredEx [2] is a credential exchange system. The users are responsible for mappings between the credentials they posses, which we believe is not a good idea. Another problem is that choosing the least privilege credential for a given operation would involve multiple calls to the service and would require the user to knowing which aliases to retrieve.

Securely Available Credentials Protocol (SACRED) [5] is another password-based credential storage and retrieval system. It doesn't perform any form of delegation and it is centralized repository as such it has the same problems as MyProxy. Also, its operation is transparent inside one Security/PKI/CA domain (i.e., doesn’t solve inter-domain use of credentials). Moreover, it disallows X.509 proxy delegation as defined in Grid Security Infrastructure (GSI) [9].

In this paper, we present the architecture of an online grid credential management service called SafeBox that facilitates the secure storage of credentials and enables the dynamic exchange of different credential types. In summary, existing credential management tools do not scale and not fault-tolerant. Thus, it is unfeasible to use them in a production grid, with a potentially very large number of users (e.g. exceeding thousands of people). Moreover, being able to exchange multiple credentials is extremely important [2]. However, dealing with several credentials does require some intelligence on the part of the credential requester (either the user or the service on the user's behalf) [2]. For instance, choosing the least privileged credential for a given operation would require knowing the privilege levels corresponding to various credential aliases [2]. Similarly, obtaining a set of credentials to present for authentication would involve several calls to the service and would require knowing which particular credentials to retrieve. As a result, most existing credential exchange services only support a single direction of credential exchange (i.e. only password for certificate, not the reverse), and tend to be limited to specific application domains (e.g., grid services) using proprietary protocols on a single platform [2].

The rest of the paper is organized as follows. Section 2 discusses the overall architecture of the SafeBox credential management system. Section 3 describes the credential management policy. Section 4 describes the credential management services. Section 5 describes the credential replication service. Section 6 describes the accounting and auditing service. Section 7 discusses the credential exchange service of the SafeBox in detail. The conclusions and future directions are discussed in Section 8.

2. SafeBox Architecture

SafeBox allows long-term credentials to be stored in a trusted server from which certificates can be obtained by providing a password. A SafeBox server is typically run by Grid administrators on a well-secured machine, to provide maximum protection for the stored credentials.

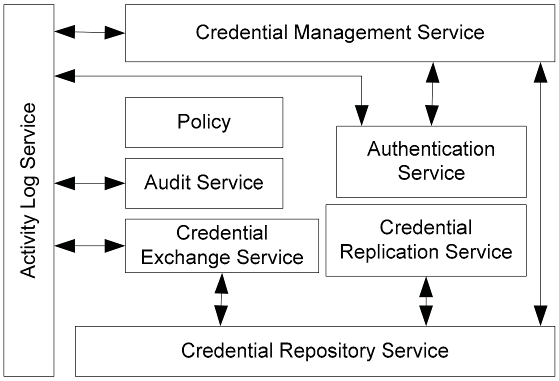
Activity Log Service
Credential Management Service
Policy
Authentication Service
Audit Service
Credential Exchange Service
Credential Replication Service
Credential Repository Service

Fig. 1: SafeBox Architecture

Fig. 1 shows an overview of the SafeBox credential management system architecture and how it interacts with various components of the Grids. The main goal of SafeBox is to get credential management out of end users hands while enable users to have access to them from anywhere at anytime. SafeBox stores encrypted credentials and access policy in an online repository and make it available to users on demand with appropriate authentication at anytime from anywhere. There are no long-term credentials for the user to manage in this case, so there are no private keys stored on potentially insecure end-user devices such as laptops.

In addition, SafeBox can perform single sign-on as well as credential delegation and provides auditing and accountability of credential use.

SafeBox could be deployed to manage the credentials of a single user, a small group within a single organization, or a large collaborative group that spans organizations. The model supported by SafeBox has advantages for the mobile user but requires very careful management of the server.

As shown in Fig. 1, SafeBox consists of five main components: (i) authentication service; (ii) credential management services (CMS); (iii) credential replication service (CRS); (iv) Audit service, and (v) credential exchange service (CES). These components collectively allow Grid users to securely store and manage multiple sets of different type credentials; allow them the flexibility to conveniently use grid services and provide flexible authentication to a wide range of systems even when away from their desktop.

In the following subsections, we will briefly explain each of these components.

3. Credential Management Policy

Access to SafeBox is governed by a server-side and client-side policies. Server-wide policies allow the SafeBox administrator to control how the SafeBox may be used. Client-side policies allow users to specify how each credential may be accessed (e.g. credential renewal policy).

A SafeBox server is located at each site or a group of sites can have one SafeBox server. Users run SafeBox client software. All communication between SafeBox clients and SafeBox servers is encrypted via the SSL protocol [6]. The server-side policy governs issues such as if the third party could upload new credentials, delete credential, ask for extension of credential lifetime, etc. The policy also determines who has access to view and modify the logs.

The default credential is the same from which the user initiates the request. In case there are multiple credentials used in the initiating site, the most often used credentials will be used as default credential. The credential exchange service can be invoked either by the user or the services.

4. Credential Management Service

SafeBox credential management service provides users with mechanism to store, retrieve, renew and revoke credentials as well as reset passwords. These operations are performed only after the client and server mutually authenticated each other. The authentication information in this process consists of a user identity and a pass phrase to be used to authenticate any later retrieval operations.

Each client of a SafeBox could own a safe box of a certain size. The safe box could expand or shrink dynamically based on the number of credentials stored in it. Each user controls the safe box under his or her username and initial password.

SafeBox allows users to store multiple credentials they own in their personal safe box. Credentials can be uploaded by the user to their own safe box. Each credential uploaded on SafeBox server has associated service name (e.g., Gmail). By allowing the end-users to give a service name to the stored credentials, the users gains the ability to store multiple different credentials which can be selectively retrieved on-demand at anytime from anywhere.

The credentials may be stored encrypted with the user's passphrase in the repository, so a compromise of the SafeBox server would not immediately compromise all user keys; an additional offline attack on the keys would be required. Both the user identity and pass phrase are chosen by the user, but can be tested by the SafeBox to ensure that they meet any local policy (e.g. the pass phrase must be a certain length, etc.).

Once credentials are stored in a SafeBox, clients can query the server for the credentials they have stored and retrieve specific credentials by name. Users could also retrieve a proxy-credential. A proxy credential is a short-term credential that is created by a user, which can be used in place of the long-term credential to authenticate that user. The proxy credential has its own private key and certificate, and is signed using the user's long-term credential. Last but not least, SafeBox supports automatic negotiation of required credentials to allow applications to retrieve the needed credentials as well as notify the user that a needed credential is unavailable.

SafeBox securely stores the user's credentials and generates proxy credentials for the user via delegation when requested. The retrieval restrictions are currently limited to a maximum lifetime for proxy credentials that the repository may delegate on the user's behalf. The credentials delegated to the repository normally have a lifetime of a week. The user can change this to any length of time desired. The user can also, at any point, destroy any credentials they previously delegated to the repository.

When a user wishes to retrieve a stored credential, a username-password pair (established during storage) and challenge-response is used to authenticate the request and if successful the required credential would then be exchanged for a proxy certificate. A user could also retrieve multiple credentials in a single request (especially when more than two tokens are required). This capability reduces the number of request messages as such can greatly improve efficiency.

5. Fault-tolerance Service

In contrast to a locally stored private key, a remote credential service is vulnerable to network and server outages. If the server is unavailable, users will either resort to other, potentially less secure, credential access mechanisms or be denied access to grid services. Therefore, fault-tolerance is an important property of any credential management service.

SafeBox provides fault-tolerance by replicating credentials over multiple SafeBox in different sites. The client machine can also be an option provided that it has tools such as Java Keystore [10]. Replication of credentials is optional and the user decided whether or not his/her credential is replicated. Also, the user is informed the secondary location of the credentials if replication is allowed.

6. Audit Service

The main goal of auditing service is to effectively report on user accesses and how credentials are used. The audit service is responsible for producing records, which track security relevant events. The auditing service captures information regarding how credentials were used within the Grid. It registers in the activity log all operations, which include information about the time and source (e.g., IP address) of the operation, the claimed identity of the client, the type and result of the operation and possibly other operation specific information.

The resulting audit records may be examined as to determine if the desired security policy is being enforced as well as adherence to the stated access control and authentication policies. Audit trails establish which identities were active and what access they had at what time. SafeBox digitally signs logs so that some level of assurance can be provided that they have not been tampered with.

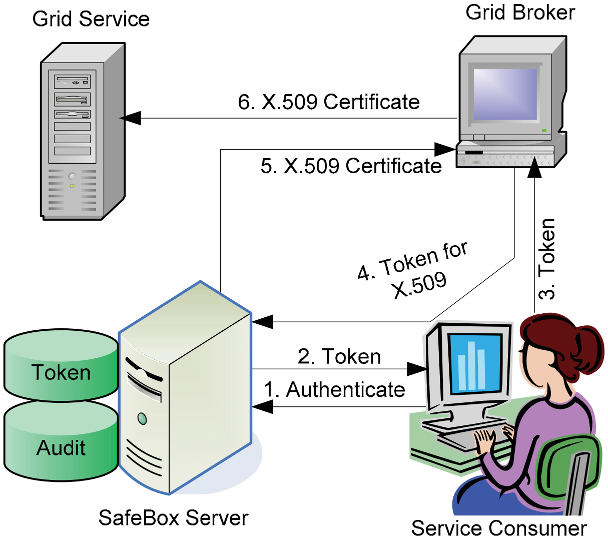

Fig. 2: Credential exchange protocol

7. Credential Exchange Service

Authentication mechanisms employed by Grid sites vary with regard to the type of security credentials used to prove user identity. SafeBox's credential translation service (CES) is an inter-domain credential manager. It provides credential translation facility between one type of credential to another type or form of credential.

In light of the InterGrid computing, being able to exchange multiple credentials in a flexible way is paramount. Fig. 2 demonstrates how CES provides automatic and flexible credential exchange services. In the scenario of Fig. 2, the SafeBox clients simply submit a job execution request along the default credentials to the Grid Resource Broker (GRB), which in turn recruits an execution site for the job.

A service consumer would like to use a grid service discovered through the Grid Broker service. The consumer communicates with the SafeBox server to obtain a security token. After authenticating the user, the SafeBox server generates a security token and sends it to the consumer, which in turn forwards the token to the Grid Broker service. When the client generates the new token in Step 2, the client is free to set the lifetime of this certificate to meet its needs. In this way, the client effectively has control over how long the delegated credential is stored to the service.

The Broker already knows the authentication credential needed by the Grid service. Since the default credential presented by the user differs from the execution site, GRB asks the SafeBox to exchange the default token with the appropriate credential (in the example, to the X.509 certificate).

The SafeBox server checks the validity of the security token and then forewords a proxy X.509 certificate [11] to the Broker. Proxy certificates are the standard authentication token used for grid services; however, it would not require a significant modification to the credential exchanger for it to support issuance of standard X.509 certificates (in which case it could function as a kind of online certification authority). Finally, the Broker will send the service request from the customer along the proxy certificate to the Grid service. The SafeBox server records all interactions in an audit database. The GRB then submits the job along the proxy certificate to the execution site for execution.

At this point, we wish to highlight some of the salient differences between the SafeBox credential exchange service described in this section with the CredEx [2]. SafeBox stores long-term user credentials and generates a proxy certificate each time a request from authorised user is received. This makes sure that the long-term credential will not be easily compromised. In contrast, the X.509 certificates stored and retrieved in CredEx are the X.509 proxy certificates [11]. This means, users must manage their long-term credentials or have another tool such as MyProxy [3] to manage it for them. Note also that CredEx have focused on identity certificates for use in authentication; it's not clear if or how exchanges involving attribute certificates might work.

In addition, the user in SafeBox does not really need to remember the type of the credentials to be exchanged as in CredEx. Instead, the user will either ask for a security token (default) or provide to the SafeBox server the service name (e.g., gmail) to retrieve the needed credential (e.g., username-password pair). In contrast, CredEx expects users to store and manage their credentials. Last but not least, the resources in Grids are acquired and released continuously at different sites so it is not possible for a user to be involved in all such transactions. Thus, user managed credentials can pose significant problems to security of the credentials and can undermine the security of the system.

Not only SafeBox is capable of dynamic user-driven or service-driven exchange of different credential types, a user can also accomplish single sign-on by acquiring a single (default) credential as shown in Fig. 2 and then dynamically exchanging that credential as needed for services that authenticate a different way.

8. Conclusions and Future Directions

This paper described a novel credential management system in web and grid service environments called SafeBox. It enables users to securely store multiple credentials and exchange different credential types transparently on demand. The credentials are maintained in a secure format by encrypting it before storage in SafeBox.

We are currently implementing the SafeBox credential management system for InterGrid environment. There are several issues that we need to add to SafeBox. For example, even though storing copies of multiple credential has it's own security problems, we are working on how to solve this problem. Also, SafeBox can be configured to allow users to logon with existing site credentials, using Pluggable Authentication Modules (PAM) and/or the Simple Authentication and Security Layer (SASL). Through these mechanisms, users are not required to remember another username and password for the SafeBox service.

Acknowledgements

This research is supported by ARC. Also, without the help of Maliha Omer, this work would not have been possible.

References

[1] Marcos Dias de Assunção, Rajkumar Buyya and Srikumar Venugopal, InterGrid: A Case for Internetworking Islands of Grids, Concurrency and Computation: Practice and Experience, 20(8), pp: 997-1024, 2008.

[2] David Del Vecchio and Marty Humphrey and Jim Basney and Nataraj Nagaratnam, CredEx: User-Centric Credential Management for Grid and Web Services, Proceedings of IEEE International Conference on Web Services, pp:149 - 156, 2005.

[3] J. Basney and W. Yurcik and R. Bonilla and A. Slagell, The Credential Wallet: A Classification of Credential Repositories Highlighting MyProxy, Proceedings of the 31st Research Conference on Communication, Information and Internet Policy, 2003

[4] Markus Lorch and Jim Basney and Dennis Kafura, A Hardware-secured Credential Repository for Grid PKIs, Proceedings of 4th ACM/IEEE Int. Symposium on Cluster Computing and the Grid, pp:, 2004.

[5] S. Farrell}, Securely Available Credentials Protocol (RFC 3767)}, IETF Network Working Group, 2004.

[6] Eric Rescorla, SSL and TLS: Designing and Building Secure Systems, Addison-Wesley Pub Co., ISBN = 0-201-61598-3.

[7] S. Tuecke and D. Engert and M. Thompson, Internet X.509 public key infrastructure proxy certificate profile, Internet Draft draft-ietf-pkix-impersonation-00.txt.}, 2001.

[8] Jemal H. Abawajy, Grid Accounting Service Infrastructure for Service-Oriented Grid Computing Systems, Lecture Notes in Computer Science, Vol. 3458, pp:168-175, 2005.

[9] Ian T. Foster and Carl Kesselman and Gene Tsudik and Steven Tuecke, A Security Architecture for Computational Grids, Proceedings of 5thACM Conference on Computer and Communications Security, pp: 83-92, 1998.

[10] keytool - Key and Certificate Management Tool. J2SE SDK Docs, accessed 5 August 2004. http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/keytool.html.

[11] S. Tuecke, et al. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile (RFC 3820). IETF Network WG, June 2004. http://www.ietf.org/rfc/ rfc3820.txt.

[12] J. Novotny and S. Tuecke and V. Welch, An Online Credential Repository for the Grid: MyProxy, Proceedings of the 10th IEEE Symposium On High Performance Distributed Computing, 2001.

[13] Jemal H. Abawajy, Grid Accounting Service Infrastructure for Service-Oriented Grid Computing Systems, Lecture Notes in Computer Science, pp: 168-175, 2005.

[14] Olga Kornievskaia and Peter Honeyman and Bill Doster and Kevin Coffman, Kerberized Credential Translation: A Solution to Web Access Control, Proceedings of the USENIX Security Symposium, 2001.

[15] F. Siebenlist, et al. OGSA Security Roadmap: Towards a Secure OGSA. July 2002. http://www.globus.org/ogsa/ security/draft-ggf-ogsa-sec-roadmap-01.doc