J. Basney
NCSA

October 2001

GSI Online Credential Retrieval - Requirements

Status of this Memo

Abstract

This memo defines requirements for online credential retrieval services that provide secure access to X.509 proxy credentials in the Grid Security Infrastructure (GSI).

Table of Contents

1 Introduction

An online credential retrieval service gives users secure and convenient access to the credentials they need for authentication. To make credentials available, the service either stores the credentials in a secure repository or generates new credentials on request.

Requiring users to manage their credentials has a number of drawbacks. First, users may not be able or willing to effectively protect their private keys from compromise or loss. Second, users may access secure services from many devices, and distributing their private keys to each device can be inconvenient and potentially insecure. Third, users may need multiple credentials to access different secure services because of differing trust policies, further increasing the user's key management burden. A service that securely manages user's credentials can therefore potentially improve both security and usability.

This memo describes usage scenarios and defines requirements for an online credential retrieval service for the Grid Security Infrastructure (GSI). We invite comments on the scenarios and requirements in this memo and suggestions for additional scenarios and requirements that should be considered. Please send comments and suggestions to the GSI working group of the Global Grid Forum by electronic mail to security-wg@gridforum.org.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [4].

2 Grid Security Infrastructure and Proxy Credentials

The term "the Grid" was coined in the mid 1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [8, 11]. The Grid Security Infrastructure (GSI) was subsequently developed, based on existing standards, to address the unique security requirements that arise in Grid environments, as described in [5, 7]. The formation of the Grid Forum as a body to promote and develop Grid technologies, and the broad acceptance that GSI has received within the Grid community, led to the formation of the Grid Forum GSI working group to foster the specification and development of GSI.

Authentication in GSI is based on proxy credentials. A proxy credential consists of a proxy certificate and an associated private key. The proxy certificate is an X.509 certificate that is

derived from a standard X.509 end entity certificate or another proxy certificate and signed by the private key associated with the source certificate. Proxy credentials serve to limit the vulnerability of the end entity private key while supporting GSI requirements for single sign-on and credential delegation.

Rather than entering a passphrase to decrypt the private key on every authentication operation, or stashing the passphrase or unencrypted key on the local system for repeated use, the user can use the private key once to create a proxy credential. The proxy certificate contains restrictions, such as a short lifetime, that limits the vulnerability if the proxy key should be compromised. The proxy key can then be stashed unencrypted for the duration of the user's session.

Proxy credentials can also be used to delegate credentials to processes acting on the end entity's behalf without transferring the end entity's private key to the process. Instead, the process generates its own proxy certificate and key and asks the delegating entity to sign the certificate, thereby allowing credentials to be forwarded over the network without transferring private keys. The delegating entity will typically place restrictions in the proxy certificate to limit the vulnerability of the delegated credential.

3 Usage Scenarios

The following usage scenarios SHOULD be supported by the online credential retrieval service.

3.1 Credential Initialization

A user logs in to a computer where his or her X.509 credentials are not locally available. The user downloads and installs a credential retrieval client program from a trusted source if one is not already installed or asks the local administrator to perform the installation. The user runs the credential retrieval client program, entering a username and passphrase, to obtain an X.509 proxy credential.

3.2 Credential Renewal

A user or service with an existing X.509 proxy credential nearing expiration runs a credential retrieval client program, authenticates to the online credential retrieval service with the existing X.509 proxy credential, and retrieves a new proxy credential with an extended lifetime.

3.3 Transparent Credential Retrieval

The user or administrator modifies the login script for the user's local account to run a credential retrieval client program that authenticates to the online credential retrieval service using the local security context, such as a Kerberos ticket, to retrieve an X.509 proxy credential transparently on each login.

3.4 Adding Delegation to Existing Protocols

In this scenario, a user accesses a Grid service using a client program that does not support credential delegation, for example, using a web browser to access a Grid Portal. The user connects to the portal, and the portal prompts the user for his or her credential information. The user enters the username and passphrase under which the user's credential can be retrieved from a credential retrieval service. The portal software runs a credential retrieval client program, using the user's username and passphrase to obtain credentials for the user, thereby allowing the portal to access Grid resources on the user's behalf.

3.5 Multiple Credentials

A user has different credentials for authenticating to Grid resources in different administrative domains. To access a Grid resource, the user's client program queries the resource for the types of credentials it is willing to accept. The client then queries the credential retrieval service to find a credential for the user that meets the Grid resource's requirements and retrieves the credential if one is found.

4 Requirements

This section lists requirements for protocols, servers, and credential repositories that provide credential retrieval services.

4.1 Protocol Requirements

The online credential retrieval service MUST support a standard protocol for retrieving credentials. Additional protocols MAY be supported to allow users and administrators to add, remove, and modify the credentials that may be retrieved from the service. A credential upload protocol allows authorized clients to insert credentials into a repository for later retrieval or remove existing credentials from a repository. Administrative protocols allow authorized clients to modify the authorization requirements for retrieving a credential and other policy restrictions on the credentials.

Each protocol SHOULD share message formats and authentication mechanisms where possible.

4.1.1 Credential Retrieval Protocol Requirements

The protocol MUST support delegation of X.509 proxy credentials from the server to the client. Retrieval of other types of credentials, including X.509 end entity credentials, is not considered at this time and is not required for the usage scenarios described above.

The protocol MUST authenticate the client to the server and MUST allow support of different client authentication mechanisms. Support for username/passphrase and X.509 authentication is REQUIRED. Additional authentication mechanisms, such as Kerberos, MAY be supported.

The protocol MUST ensure the integrity of the client's authenticated credential retrieval request (i.e., using a message integrity check).

If the retrieval protocol requires the client to transfer a secret, such as a passphrase, to the server, the protocol MUST authenticate the server to the client before transferring the secret and the secret MUST be encrypted in transit.

The client MUST verify that the retrieved credentials contain the expected attributes.

The protocol SHOULD support replication of the retrieval service, and the protocol definition SHOULD include a method for locating a credential retrieval server that can provide the credentials requested by the client. The credential service may be partially replicated, so a given credential may be available from some but not all servers. A credential tag and server/domain name may be required for retrieval requests.

The protocol SHOULD allow the client to choose the attributes of the credential to be retrieved (according to what the server will allow). For example, the client may be authorized to obtain credentials signed by different certificate authorities, possibly with different subjects, or the client may request that the server delegate a credential with specified restrictions. Some mechanism for querying the set of available credentials would be needed to support this functionality.

4.1.2 Credential Upload Protocol Requirements

Note: This protocol would apply only to credential retrieval systems that use a credential repository. The set of credentials available from online certificate authorities, which generate credentials on demand, would be controlled by an administrative protocol rather than an upload protocol.

The protocol MUST support delegation of X.509 proxy credentials from the client to the server.

The protocol MUST allow authenticated clients to remove previously delegated credentials from the repository.

The protocol MUST allow the client to associate one or more authentication requirements with an uploaded credential. The client can choose the username/passphrase pair(s) or X.509 identities that are authorized to retrieve the credential.

The protocol MUST allow the client to specify lifetime restrictions for retrieved credentials that are shorter than the lifetime of the uploaded credential. This allows the client to upload a long-lived credential to the repository while minimizing the vulnerability of credentials retrieved from the repository.

The protocol MUST authenticate the server to the client to prevent uploading credentials to an untrusted server.

The protocol SHOULD authenticate the client to the server and verify that the client is authorized to upload credentials. Client authentication may not be needed for "public utility" servers willing to store credentials for any Grid users.

The protocol MAY allow the client to associate additional restrictions with the credential to be enforced by the server beyond any policy restrictions encoded in the credential itself.

4.1.3 Administrative Protocol Requirements

The protocol SHOULD allow authorized clients to associate new authentication requirements for retrieval of credentials. For example, clients can associate a new username/passphrase with a credential.

4.2 Credential Server Requirements

The server MUST restrict authenticated clients to retrieve only those credentials for which they are authorized. The server SHOULD allow multiple (identity, authentication mechanism) pairs to be authorized to retrieve credentials, on a per-credential basis.

The server MUST enforce limits on the maximum lifetime of delegated credentials, both on a per-credential basis and for all credentials managed by the server.

The server SHOULD securely log all protocol transactions for auditing purposes.

The server MAY support online notification of protocol transactions to authorized parties, including notification of requests that must be authorized before they proceed.

4.3 Credential Repository Requirements

Private keys stored in the repository SHOULD be encrypted and the information required to decrypt the keys SHOULD NOT be stored in the repository. In this case, the client must include the information required to decrypt the key in the credential retrieval request, so the server can decrypt the key and use it to perform delegation. However, the server should discard the decrypted key and the information used to decrypt it immediately after performing the delegation. This may not be possible for all authentication mechanisms. For passphrase-based authentication, the private keys can be encrypted with the passphrase.

The credential repository SHOULD be replicable.

5 Related Work

Protocols for secure credential retrieval are under development in the IETF Securely Available Credentials (SACRED) working group. The working group has produced a requirements document [2] and draft framework and protocol documents. Many of the SACRED requirements are equivalent to requirements listed in this memo. However, the SACRED requirements state that the credential format MUST be opaque to the protocol and the protocol MUST NOT force credentials to be present in cleartext at the server. These requirements disallow X.509 proxy delegation as specified in this memo. The author(s) of this memo will work with the SACRED working group to address this issue. The development of standards for online credential retrieval

in GSI SHOULD include input from the SACRED working group and SHOULD be compatible when possible with SACRED requirements and standards.

The IETF Public-Key Infrastructure (X.509) (PKIX) working group has developed standard protocols for the online management of credentials. In particular, [1, 3, 9] define protocols for retrieving certificates and certificate revocation lists from an online repository and [10] defines an online certificate status protocol. These protocols obtain public PKI information from a repository, whereas an online credential retrieval protocol involves private keys that must be kept secret. Despite this difference, an online credential retrieval protocol SHOULD be compatible where possible with the PKI certificate management protocol framework.

6 Security Considerations

Centralized credential management raises significant security concerns. The central server is an attractive target for attack because of the large number of credentials that may be compromised.

The compromise of a certificate repository could potentially compromise all credentials stored there. Encrypting the credentials as recommended above can limit the vulnerability by requiring an additional offline attack to decrypt the credentials. However, a compromised server could instead wait for clients to retrieve their credentials to learn the encryption keys.

The compromise of a server that generates credentials on demand by signing them with a certificate authority key (i.e., an online certificate authority) would allow an attacker to generate and use credentials for any principal in the security domain until the certificate authority key is revoked.

It is important to place these risks in context. Kerberos Key Distribution Centers implement a form of centralized credential management, so community experience with Kerberos security can suggest best practices for securing other types of centralized credential servers. A professionally administered, dedicated credential server should provide a higher level of security than current practice, where end users store their private keys on less secure end systems, including network file systems where eavesdropping on unencrypted traffic is possible.

A credential retrieval service deployed by an organization must be acceptable under the relevant certificate authority policy documents [6]. The management practices for deployed credential retrieval systems should be documented and audited.

Keys used by the credential retrieval service and credentials retrieved from the service are not suitable for generating non-reputable digital signatures because the credential retrieval service has access to the keys. However, the suitability of end entity credentials for generating non-reputable signatures is not affected by delegating proxy credentials to a credential retrieval service. The credential retrieval service is not intended to manage keys for digital signatures and is not a key escrow system.

7 Acknowledgements

Discussions with Randy Butler, Laura Pearlman, Steve Tuecke, and Von Welch led to the initial version of this memo. The paragraph introducing the Grid, GSI, and the Grid Forum was taken from a memo written by Steve Tuecke.

8 References

[1] Adams, C. and S. Ferrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC 2510, March 1999.

[2] Arsenault, A. and S. Ferrell, "Securely Available Credentials - Requirements," RFC 3157 (Informational), August 2001.

[3] Boeyen, S., T. Howes, and P. Richard, "Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2," RFC 2559, April 1999.

[4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[5] Butler, R., D. Engert, I. Foster, C. Kesselman, and S. Tuecke, "A National-Scale Authentication Infrastructure," IEEE Computer, vol. 33, pp. 60-66, 2000.

[6] Chokhani, S. and W. Ford, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," RFC 2527 (Informational), March 1999.

[7] Foster, I., C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," presented at Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998.

[8] Foster, I., C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of Supercomputer Applications, 2001.

[9] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP," RFC 2585, May 1999.

[10] Myers, M., R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," RFC 2560, June 1999.

[11] Stevens, R. et. al., "From the I-WAY to the National Technology Grid," Communications of the ACM, vol. 40, pp. 50-61, 1997.

9 Contact Information

Jim Basney
Alliance Computational Environments and Security Division
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
605 E. Springfield Ave.

Champaign, IL 61820-5518
Phone: 217-244-1954
Email: jbasney@ncsa.uiuc.edu