INTERNET-DRAFT   Nicolas Williams
Sun Microsystems
November 2004

GSS-APIv2 Extension for Storing Delegated Credentials

<draft-williams-gssapi-cred-store-00.txt>

Status of this Memo

Copyright Notice

Abstract

The details of Generic Security Service (GSS) credential store management vary by platform and even by GSS mechanism. Credential store management is an interesting concept that requires exploration.

This document defines a small extension to the GSS-API for GSS-API credential store management. While exploration of the credential store management problem is the goal of this document, implementation of these interfaces is not discounted nor discouraged.

Conventions used in this document

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 [RFC2119].

Table of Contents

1. Introduction

[Text needed on what is a "credential store" and what is a "current credential store,: and their relation to the callers' current execution context.]

[Also add text about how this stuff imports concepts such as "process," which does not augur well for interface genericity.]

[See [gss_store_cred].]

2. GSS_Make_cred_store()

Inputs:

```
o inheritance SET OF ENUMERATED,  -- Specifies the desired
-- inheritance rule for this store.  Possible values include:
--
--  o none (this process only)
--  o default
--  o spawn
--  o fork
--  o exec

o sharing ENUMERATED,  -- Specifies the desired degree of sharing
-- of this store with other processes or threads.  Possible values
-- include:
--
--  o none
--  o default
--  o allThreadsInSameProcess
--  o allProcessesInSameSession
--  o allProcessesForSameUser
--  o allProcesses
```

Outputs:

```
o major_status INTEGER,

o minor_status INTEGER,

o cred_store_handle CREDENTIAL STORE HANDLE
```

Return status codes:

...

3. GSS_Get_current_cred_store()

Inputs:

o <none>

Outputs:

o major_status INTEGER,

o minor_status INTEGER,

o cred_store_handle CREDENTIAL STORE HANDLE

Return status codes:

o GSS_S_COMPLETE indicates that there is a credential store or that one can be created, when GSS_Store_cred() is called, for the current execution context of the caller.

o GSS_S_UNAVAILABLE indicates that no credential store exists for the current execution context of the caller.

o GSS_S_FAILURE indicates that an unspecified failure has occurred.

This function returns a credential store handle that refers to the credential store from which credentials would be acquired given the current execution context of the caller.

Credential store handles may not remain accessible when the caller switches the user of the execution context.

4. GSS_Set_current_cred_store()

Inputs:

o cred_store_handle CREDENTIAL STORE HANDLE,

Outputs:

o major_status INTEGER,

o minor_status INTEGER

Return status codes:

o GSS_S_COMPLETE indicates that the given credential store will be used by subsequent GSS-API credential acquisition or storage made in the same execution context as that of the caller to GSS_Set_current_cred_store(). If the given store handle is GSS_C_NO_STORE then either a default or new (which is a platform-specific matter) credential store will be created and set as the current credential store.

- GSS_S_BAD_STORE indicates that the given credential store handle is not recognized or refers to a credential store that no longer exists or is otherwise corrupt.

o GSS_S_UNAVAILABLE indicates that the current credential store for the current execution context could not be set, possibly due to lack of resources.

o GSS_S_FAILURE indicates that a generic failure has occurred.

This function changes the credential store for the current execution context.

Calls to this function MAY have platform-specific side effects (e.g., setting environment variables, setting a process' "pag," etc...), but an implementation of it MUST NOT change the user context of the application, a restriction applicable only on multi-user platforms.

The current credential store may change or become unavailable when the caller switches the user of the execution context.

5. GSS_Inquire_cred_store()

[Inquire a cred store for inheritance and sharing levels, supported mechanisms.]

6. GSS_Display_cred_store()

[Display a credential store. A generic equivalent of MIT's klist(1).]

7. C-Bindings

[...]

8. Examples

[...]

9. Security Considerations

10. Acknowledgements

[...]

11. References

11.1. Informative References

[gss_store_cred]
N. Williams, draft-williams-gssapi-store-deleg-creds-00: "GSS-APIv2 Extension for Storing Delegated Credentials," September 2003, Status: Internet-Draft.

11.2.   Normative References

[RFC2026]
S. Bradner, RFC2026: "The Internet Standard Process - Revision 3," October 1996, Obsoletes - RFC 1602, Status: Best Current Practice.

[RFC2119]
S. Bradner, RFC2119 (BCP14): "Key words for use in RFCs to Indicate Requirement Levels," March 1997, Status: Best Current Practice.

[RFC2743]
J. Linn, RFC2743: "Generic Security Service Application Program Interface Version 2, Update 1," January 2000, Status: Proposed Standard.

[RFC2744]
J. Wray, RFC2744: "Generic Security Service API Version 2 : C-bindings," January 2000, Status: Proposed Standard.

12. Author's Address

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
Email: Nicolas.Williams@sun.com

Full Copyright Statement

TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.