Internet Engineering Task Force (IETF)
Request for Comments: 5896
Updates: 4120
Category: Standards Track
ISSN: 2070-1721

L. Hornquist Astrand
Apple, Inc.
S. Hartman
Painless Security, LLC

June 2010

Generic Security Service Application Program Interface (GSS-API): Delegate if Approved by Policy

Abstract

Several Generic Security Service Application Program Interface (GSS-API) applications work in a multi-tiered architecture, where the server takes advantage of delegated user credentials to act on behalf of the user and contact additional servers. In effect, the server acts as an agent on behalf of the user. Examples include web applications that need to access e-mail or file servers, including CIFS (Common Internet File System) file servers. However, delegating the user credentials to a party who is not sufficiently trusted is problematic from a security standpoint. Kerberos provides a flag called OK-AS-DELEGATE that allows the administrator of a Kerberos realm to communicate that a particular service is trusted for delegation. This specification adds support for this flag and similar facilities in other authentication mechanisms to GSS-API (RFC 2743).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5896.

Table of Contents

1. Introduction

Several GSS-API applications work in a multi-tiered architecture, where the server takes advantage of delegated user credentials to act on behalf of the user and contact additional servers. In effect, the server acts as an agent on behalf of the user. Examples include web applications that need to access e-mail or file servers, including CIFS file servers. However, delegating user credentials to a party who is not sufficiently trusted is problematic from a security standpoint.

Today, GSS-API [RFC2743] leaves the determination of whether delegation is desired to the client application. An application requests delegation by setting the deleg_req_flag when calling init_sec_context. This requires client applications to know what services should be trusted for delegation.

However, blindly delegating to services for applications that do not need delegation is problematic. In some cases, a central authority is in a better position than the client application to know what services should receive delegation. Some GSS-API mechanisms have a

facility to allow an administrator to communicate that a particular service is an appropriate target for delegation. For example, a Kerberos [RFC4121] KDC can set the OK-AS-DELEGATE flag in issued tickets as such an indication. It is desirable to expose this knowledge to the GSS-API client so the client can request delegation if and only if central policy recommends delegation to the given service.

This specification adds a new input flag to gss_init_sec_context() to request delegation when approved by central policy. In addition, a constant value to be used in the GSS-API C bindings [RFC2744] is defined. Finally, the behavior for the Kerberos mechanism [RFC4121] is specified.

2. Requirements Notation

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].

3. GSS-API flag, C binding

The gss_init_sec_context API is extended to gain a new input flag, deleg_policy_req_flag, and a new output flag, deleg_policy_state BOOLEAN. If the deleg_policy_req_flag is set, then delegation SHOULD be performed if recommended by central policy. When delegation was recommended by the central policy and when delegation was done, the output flag deleg_policy_state will be set.

In addition, the C bindings are extended to define the following constant to represent both deleg_policy_req_flag and deleg_policy_state (just like GSS_C_DELEG_FLAG maps to two flags).

```
#define GSS_C_DELEG_POLICY_FLAG 32768
```

4. GSS-API Behavior

As before, if the deleg_req_flag is set, the GSS-API mechanism will attempt delegation of user credentials. When delegation is successful, deleg_state will return TRUE in both the initiator and acceptor output state (gss_init_sec_context and gss_accept_sec_context, respectively).

Similarly, if the deleg_policy_req_flag is set, then the GSS-API mechanism will attempt delegation if the mechanism-specific policy recommends to do so. When delegation is allowed and successful, deleg_state will return TRUE in both initiator and acceptor output

state. In addition, deleg_policy_state will be set in the initiator output state.

If the initiator sets both the deleg_req_flag and deleg_policy_req_flag, delegation will be attempted unconditionally. When delegation is successful, deleg_state will return TRUE in the initiator and acceptor. When delegation was successful, the deleg_state will return TRUE in the initiator and acceptor. Additionally, if the mechanism-specific policy recommended delegation, the deleg_policy_state will additionally return TRUE for the initiator (only).

Note that deleg_policy_req_flag and deleg_policy_state apply the initiator only. Their state is never sent over the wire.

5. Kerberos GSS-API Behavior

If the initiator sets the deleg_policy_req_flag (and not deleg_req_flag), the Kerberos GSS-API mechanism MUST only delegate if OK-AS-DELEGATE is set [RFC4120] in the service ticket. Other policy checks MAY be applied. If the initiator sets deleg_req_flag (and not deleg_policy_req_flag), the behavior will be as defined by [RFC2743]. If the initiator set both the deleg_req_flag and deleg_policy_req_flag, delegation will be attempted unconditionally.

[RFC4120] does not adequately describe the behavior of the OK-AS-DELEGATE flag in a cross realm environment. This document clarifies that behavior. If the initiator sets the deleg_policy_req_flag, the GSS-API Kerberos mechanism MUST examine the OK-AS-DELEGATE flag in the service ticket, and it MUST examine all cross realm tickets in the traversal from the user's initial ticket-granting-ticket (TGT) to the service ticket. If any of the intermediate cross realm TGTs do not have the OK-AS-DELEGATE flag set, the mechanism MUST NOT delegate credentials.

6. Rationale

Strictly speaking, the deleg_req_flag behavior in [RFC2743] could be interpreted the same as deleg_policy_req_flag is described in this document. However, in practice, the new flag is required because existing applications and user expectations depend upon GSS-API mechanism implementations without the described behavior, i.e., they do not respect OK-AS-DELEGATE.

In hind sight, the deleg_req_flag should not have been implemented to mean unconditional delegation. Such promiscuous delegation reduces overall security by unnecessarily exposing user credentials, including to hosts and services that the user has no reason to trust.

Today there are Kerberos implementations that do not support the OK-AS-DELEGATE flag in the Kerberos database. If the implementation of the deleg_req_flag were changed to honor the OK-AS-DELEGATE flag, users who deploy new client software would never achieve credential delegation because the KDC would never issue a ticket with the OK-AS-DELEGATE flag set. Changing the client software behavior in this way would cause a negative user experience for those users. This is compounded by the fact that users often deploy new software without coordinating with site administrators.

7. Security Considerations

This document introduces a flag that allows the client to get help from the KDC in determining to which servers one should delegate credentials, and the servers to which the client can delegate.

The new flag deleg_policy_req_flag is not communicated over the wire, and thus does not present a new opportunity for spoofing or downgrading policy in and of itself.

Mechanisms should use a trusted/authenticated means of determining delegation policy, and it must not be spoofable on the network.

Delegating the user’s TGT is still too powerful and dangerous. Ideally, one would delegate specific service tickets, but this is out of scope of this document.

A client’s failure to specify deleg_policy_req_flag can at worst result in NOT delegating credentials. This means that the client does not expand its trust, which is generally safer than the alternative.

8. Acknowledgements

Thanks to Disco Vince Giffin, Thomas Maslen, Ken Raeburn, Martin Rex, Alexey Melnikov, Jacques Vidrine, Tom Yu, Hilarie Orman, and Shawn Emery for reviewing the document and providing suggestions for improvements.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC2744] Wray, J., "Generic Security Service API Version 2 : C-bindings", RFC 2744, January 2000.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4121] Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005.

Authors' Addresses

Love Hornquist Astrand
Apple, Inc.

EMail: lha@apple.com

Sam Hartman
Painless Security, LLC

EMail: hartmans-ietf@mit.edu