Common Authentication Technologies Next Generation (Kitten)

IETF 61

Introduction

Welcome

- Kitten has been approved by the IESG as a new working group
- Active discussions on the list
- Progress made on defining the gss-naming problem; SPNEGO; PRF; and C# bindings
- Road map draft published
- New mailing list available/archives online

Agenda

Agenda

- Introduction and Welcome [5 minutes]
- Charter Review [20 minutes]
- Pseudo-random function API [5 minutes]
- Domain Based Names [10 minutes]
- C# Bindings for GSSAPI [10 minutes]
- SPNEGO issues [10 minutes]
- GSSAPI naming [10 minutes]
- Open Floor

Charter Review

Charter (1)

- The Generic Security Services API [RFC 2743, RFC 2744] provides an API for applications to set up security contexts and to use these contexts for per-message protection services. The Common Authentication Technology Next Generation Working Group (Kitten) will work on standardizing extensions and improvements to the core GSSAPI specification and language bindings that the IETF believes are necessary based on experience using GSSAPI over the last 10 years. Extensions may be published as separate drafts or included in a GSSAPI version 3. While version 2 of the GSSAPI may be clarified, no backward incompatible changes will be made to this version of the API.

Charter (2)

- This working group is chartered to revise the GSSAPI v2 RFCs for the purpose of clarifying areas of ambiguity:
  - Use of channel bindings
  - Thread safety restrictions
  - C Language utilization (e.g., type utilization, name spaces)
  - Guidelines for GSS-API mechanism designers
  - Guidelines for GSS-API application protocol designers

Charter (3a)

- This working group is chartered to specify a non-backward compatible GSSAPI v3 to support the following extensions:
  - Clarify the portable use of channel bindings and better specify channel bindings in a language-independent manner.
  - Specify thread safety extensions to allow multi-threaded applications to use GSSAPI
  - Definitions of channel bindings for TLS, IPSec, SSH and other cryptographic channels based on work started in the NFSV4 working group.

Charter (3b)

- Define a GSSAPI extension to allow applications to store credentials.
- Extensions to solve problems posed by the Global Grid Forum's GSSAPI extensions document.
- Extensions to deal with mechanism-specific extensibility in a multi-mechanism environment.
- Extend the GSS-API to support authorization by portable GSS applications while also supporting mechanisms that do not have a single canonical name for each authentication identity.

Charter (3c)

- Specify a Domain-based GSS service principal name consisting of:
  service name, host name, and domain name for use by application services hosted across multiple servers.
- Extensions to support stackable GSSAPI mechanisms.
- Define a Psuedo-Random Function for GSSAPI

Charter (4)

- This working group is chartered to perform the following GSSAPI mechanism specification work:
  - Specify a GSSAPI v2/v3 Channel Conjunction Mechanism
  - Revise RFC 2748 (SPNEGO) to correct problems that make the specification unimplementable and to document the problems found in widely-deployed attempts to implement this spec.
  - Update the GSSAPI Java Language Bindings to match actual implementation

Charter (5)

- This working group is chartered to perform the following new GSSAPI Language Binding specification work:
  - Specify a language binding for C#

Goals and Milestones (1)

Nov 04 First Meeting

Mar 05 First drafts of either 'Clarifications to GSSAPIv2' as Informational OR submit 'Generic Security Service Application Program Interface Version 2, Update 2' and 'Generic Security Service API Version 2, Update 2 : C-bindings' to the IESG as Proposed Standard

Jul 05 Submit either 'Clarifications to GSSAPIv2' as Informational OR submit 'Generic Security Service Application Program Interface Version 2, Update 2' and 'Generic Security Service API Version 2, Update 2 : C-bindings' to the IESG as Proposed Standard

Jul 05 Submit 'The Channel Conjunction Mechanism (CCM) for the GSSAPI' to the IESG as Proposed Standard Jul 05 Submit 'On the Use of Channel Bindings to Secure Channels' to the IESG as Proposed Standard

Goals and Milestones (2)

Jul 05 Submit 'The Simple and Protected GSS-API Negotiation Mechanism (Revised)' to the IESG as Proposed Standard

Nov 05 Submit 'GSSAPI Mechanisms without a Unique Canonical Name' to the IESG as Proposed Standard

Jul 06 Submit 'Generic Security Service Application Program Interface Version 3' to the IESG as Proposed Standard

Jul 06 Submit 'Generic Security Service API Version 3 : C-bindings' to the IESG as Proposed Standard

Jul 06 Submit 'Generic Security Service API Version 3 : Java and C# bindings' to the IESG as Proposed Standard

Nov 06 Charter Review

Mailing List

General Discussion: kitten@lists.ietf.org

To Subscribe:

https://www1.ietf.org/mailman/listinfo/kitten

Archive:

http://www1.ietf.org/mail-archive/web/kitten/current/index.html

Roadmap

- draft-williams-gssapi-v3-guide-to provides a roadmap to the work being done in this group. It will be updated to reflect the current state of the work prior to each IETF meeting

Pseudo-Random Function

Nico Williams

GSS-API PRF Extension

- Some applications would like a way to get a 'key' from a GSS security context
  - Dangerous and/or breaks abstractions
  - Cipher mismatch problems
- A PRF based on a key associated with a GSS sec context doesn't suffer from those problems
- So: GSS_Pseudo_random().

GSS_Pseudo_random()

- Inputs:
  - Established sec context (not prot_ready)
  - Variable length octet string
  - Desired length of output octet string
- Outputs:
  - Major, minor status codes
  - Octet string

GSS-API PRF Properties

- Output octet string is the output of a pseudo-random function (PRF) keyed with key material from the sec context as applied to the input octet string
- Calls to GSS_Pseudo_random() by the initiator and acceptor on the same sec context with the same input results in the same output
  - Apps must be careful, use once per-sec context
- Actual PRF algorithm, keying is mech-specific

Kerberos V GSS Mech PRF

- Construct PRF+ based on Kerberos V crypto framework PRF
  - krb5_gss_prf(data, length) = truncate(length, k5prf(0 || data) || k5prf(1 || data) || .. || k5prf(n || data))
- Use acceptor subkey always for new mechanism
  - Use acceptor or, if there is none, the initiator subkey for old rfc1964 mech
- Key usage TBD
  - Actual key for prf should be derived from context key

GSS/Kerberos PRF I-D Status

- Must fold edits in for:
  - PRF+, not PRF
  - Actual PRF+ spec for the Kerberos V mechanism
- Security considerations
- Both I-Ds will soon be ready for WG Last Call
- Draft names:
  - draft-williams-gssapi-prf
  - draft-williams-krb5-gssapi-prf

Domain Based Names

Nico Williams

Domain-Based Naming

- Three-part names:

  <service>, <domain>, <host>

- Purpose:

  To allow for simple validation, by initiators, of acceptors' authority to serve domain-specific resources

- New GSS name-type OID (TBD)

  GSS_C_NT_DOMAINBASED_SERVICE

- Generic name syntax:

  <service>@<domain>@<host>

Domain-Based Naming: Uses

- LDAP
  - Make sure you're talking to an LDAP server that can serve the directory data you care for
- NFSv4
  - Finding namespace roots
- In either case the client may use DNS SRV records to find servers for some domain, but perhaps not DNSSEC
  - Domain-based naming mitigates for lack of DNSSEC

Domain-Based Naming for the Kerberos V GSS Mechanism

- Name form:

  <service>/<hostname>/<domain>@<REALM>

- Realm of domain-based service:
  - Realm of host, or realm of domain?
  - If realm of domain, how to find it?

GSS/Kerberos V Domain-Based Naming I-D Status

- Must fold edits in for:
  - Domain name not optional in 'query' name form
  - Switch order of krb5 princ name components
- Security considerations
- Both I-Ds will soon be ready for WG Last Call
- Draft names:
  - draft-williams-gssapi-domain-based-names
  - draft-williams-krb5-gssapi-domain-based-names

C# Bindings

Corby Morris
Novell

Goals for C# Bindings

- Propose a C# binding for GSSAPI that uses a published standard api. The GSSAPI Java bindings as documented in RFC 2853 was a good fit for C#. Therefore the proposal is to simply include the C# binding in this RFC.
- Document any C# language specific differences from the existing Java binding.
- Keep the C# binding as close to the existing Java binding as possible to reduce documentation, etc.

Extensions to RFC 2853

- New C# assembly namespace: org.ietf.gss
- Makes note that all exception codes remain the same as specified in the Java bindings. However, C# does not have a 'throws' statement. Therefore, method prototypes do not include the exception type. There is an example in the draft.
- Provide sample C# code.
- States that all methods, datatypes & error codes should remain the same as specified in RFC 2853.

Corrections to Java Bindings

- The Sun Microsystems Java org.ietf.jgss package is close but not exactly the same as RFC 2853.
- Open question: should a revised Java/C# Bindings RFC be updated to match the deployed API?

SPNEGO bis

Larry Zhu
Microsoft Corporation
IETF 61

Goals

- Clarifications for RFC 2478
- Backward compatible with existing Windows SPNEGO implementations
- Must be secure

Draft Status

- Initial draft published

draft-zhu-spnego-2478bis-00.txt

- Highlights:
  - Safe-to-omit-MIC rules proposed
  - Pro: secure and backward compatible when there is no interference
  - Con: can incur an extra leg if the target has policy to select a mech out of order
- Good news: fair amount of reviewers from different camps
- Not-so-great news: not enough review time

Significant Issues

- Safe-to-omit-MIC rules: what are they, and why they are secure?
- Should we protect reqFlags?
- SHOUD or MAY use the optimistic token? Close to reach consensus
- Negotiation protocol w/o integrity protection?
- How is the MIC token computed?
- Do we need out-of-band negotiation with down-level clients and servers?

Special Thanks to Reviewers

- Sam Hartman, MIT
- Wyllys Ingersoll, SUN
- Ken Raeburn, MIT
- Martin Rex, SAP
- Luke Howard, PADL.COM

Improving GSS Naming

Sam Hartman
MIT

Goals

- Support authentication of internal identities (uuids).
- Allow applications to work with parts of composite names.
- Query available credentials and presented names based on components.
- Find most appropriate credential for a target.

GSSAPI V2 Compatibility

- New mechanisms used by old applications
- File formats for ACLs containing names

Possible Solutions

- Name Attributes
- Client asserted names
- Credential extensions
- Credential enumeration

Name Attributes

- Names are composed of attributes.
- Attributes are a OID label and content.
- Add operations to query, construct and manipulate names.

Client Asserted Names

- Allow clients to assert part of a name to be exported.
- Provide better compatibility with old applications.

Credential Extensions

- Add labeled extensions to credentials.
- Similar in functionality to name attributes
- Requires more changes to contexts.

Credential Enumeration

- Facility to find credentials that are available.
- Enumerate all credentials.
- Alternatively, find all possible names.

Discussion

Discussion Items

- Should Mechanism specifications of new GSS extensions such as Krb5 PRF and Domain Based Names be done in this group?

Thank you for attending

kitten@lists.ietf.org
https://www1.ietf.org/mailman/listinfo/kitten