NETWORK WORKING GROUP N. Williams
Internet-Draft Sun
Expires: January 13, 2005 July 15, 2004

On the Use of Channel Bindings to Secure Channels
draft-ietf-nfsv4-channel-bindings-02.txt

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 13, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document defines and formalizes the concept of channel bindings to secure layers and defines the channel bindings for several types

of secure channels.

The concept of channel bindings allows applications to prove that the end-points of two secure channels at different network layers are the same by binding authentication at one channel to the session protection at the other channel. The use of channel bindings allows applications to delegate session protection to lower layers, which may significantly improve performance for some applications.

Table of Contents

1. Conventions used in this document 3
2. Introduction 4
3. Definitions 6
4. Authentication protocols and channel bindings 7
  4.1 The GSS-API and channel bindings 7
  4.2 SASL and channel bindings 7
5. Channel bindings for various secure layers 8
  5.1 Bindings to SSHv2 channels 8
  5.2 Bindings to TLS channels 8
  5.3 Bindings to IPsec 8
    5.3.1 Interfaces for creating IPsec channels 9
  5.4 Bindings to other types of channels 9
6. Benefits of channel bindings to secure channels 10
7. Security Considerations 11
8. References 12
8.1 Normative 12
8.2 Informative 12
Author's Address 12
A. Acknowledgments 13
Intellectual Property and Copyright Statements 14

1. 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].

2. Introduction

Over the years several attempts have been made to delegate session protection at one network layer to another, for performance and/or scalability as well as for design elegance and also to avoid having to reinvent the wheel (that is, cryptographic session protection) for every new application or security layer.

The critical security problem to solve in order to achieve such delegation of session protection is always the same: how to ensure that there is no man-in-the-middle (MITM), from the point of view the application, at the lower network layer to which session protection is to be delegated.

An alternative statement of the problem: how does one ensure that the end-points of two secure channels at different network layers are the same?

And there may well be a MITM, particularly if the lower network layer either provides no authentication or if there is no connection between the authentication or principals used at the application and those used at the lower network layer.

Such MITM attacks can be effected by, for example, spoofing IP address lookups (which is possible, for example, when using DNS but not DNSSEC) in a way that the application may not detect but which directs the client application or network stack to connect to a different host than had been intended (e.g., to the MITM's host).

Even if such MITM attacks seem particularly difficult to effect, the attacks must be prevented for certain applications to be able to make effective use of technologies such as IPsec.

A solution to this problem is highly desirable, particularly where multi-user applications are run over secure network layers (e.g., NFS over IPsec). For such applications the authentication model used at the application layer (usually user<->server) is generally very different from that used by secure, lower network layers, such as IPsec (usually client<->server or single-user<->server), and may even use different authentication infrastructures altogether (e.g., Kerberos V for the application layer, x.509 certificates at the lower layer). Such applications cannot, at present, generally leverage the security provided by the lower network layers, which, if they could, would allow them to offload session security to the secure lower layer.

One solution involves ensuring the use of secure name services for hostname to network address translation along with the use of secure networks (e.g., IPsec). This approach can prevent the MITM attack described above, but does not offer applications any guarantees that there is no MITM in the lower layer.

This document describes another solution: the use of "channel bindings" (a GSS-API concept [RFC2743]) to bind authentication at application layers to secure transports at lower layers in the network stack.

"Channel bindings" are data which securely identify a secure channel such that, when verified to match on both endpoints of end-to-end application connections, leave no doubt that the endpoints of two secure channels (the one identified by the bindings and the one used to exchange/verify the bindings) are the same.

Because many applications exist which provide for authentication at the application layer, because many such applications use generic authentication frameworks, such as the GSS-API and SASL and are already deployed along with a common authentication infrastructure (e.g., Kerberos V, PKI, etc...), because such applications exist which multiplex multiple users onto a single session (and so cannot leverage network [e.g., IKE] authentication), the use of channel bindings is an elegant solution even where secure name services and networks are deployed.

A formal definition of the channel bindings concept is given below, as well as the specific formulation of channel bindings for various protocols that provide for session security.

Specific instructions for the use of channel bindings with GSS-API instructions is given elsewhere.

3. Definitions

The GSS-API [RFC2743] is a generic interface to GSS-API security mechanisms which provides for authentication and session cryptographic protection. One facility provided by the GSS-API is a concept of "channel bindings" which consists of some data which must be provided, if at all, by both, initiators and acceptors, and which the GSS-API security mechanisms ensure are the same for both, the initiator and acceptor of any given GSS-API security context - if the channel bindings provided by them do not match then the mechanism fails to establish the security context.

- Channel bindings

  Generally some data which names a channel or its end-points. The security properties and channel bindings of the channel, once established, MUST NOT change for the lifetime of the channel.

  More formally, there are two types of channel bindings:

  + bindings that name a channel in a cryptographically secure manner (e.g., the session ID in SSHv2; see below);
  + bindings that name the authenticated end-points of a channel (e.g., as in IPsec; see below) which are, in turn, securely bound to the channel.

  Applications can exchange authenticated, integrity-protected verifiers of channel bindings data to prove that the end-points of some channel are the logically the same as the application endpoints and thus, there can be no MITM at the lower layer.
- Channel bindings to network addresses

  The GSS-API originally defined only channel bindings to network addresses.
  The network addresses of a channel's end-points typically say nothing about the protection afforded by that channel, and where the channel can be said to be secure the network addresses may not be securely bound to the channel anyways.
  In practice channel bindings to network addresses have mostly just caused trouble with Network Address Translation (NAT).

4. Authentication protocols and channel bindings

Some authentication services provide for channel bindings, such as the GSS-API and some GSS-API mechanisms, whereas others may not, such as SASL.

Where suitable channel bindings facilities are not provided application protocol designers may include a separate, protected (where the authentication service provides message protection services) exchange of channel bindings material.

4.1 The GSS-API and channel bindings

The GSS-API provides for the use of channel bindings during initialization of GSS-API security contexts, though GSS-API mechanisms are not required to support this facility.

This channel bindings facility is described in detail in RFC2744.

GSS-API applications must agree a priori, through negotiation or otherwise, on the use of channel bindings. This is because the GSS-API does not have a way to indicate that a security context was successfully established but that the channel bindings supplied could not be verified to be the same for both peers.

Fortunately, it is possible to design GSS-API pseudo-mechanisms that simply wrap around existing mechanisms for the purpose of allowing applications to negotiate the use of channel bindings within their existing methods for negotiating GSS-API mechanisms. For example, NFSv4 [RFC3530] provides its own GSS-API mechanism negotiation, as does the SSHv2 protocol [SECSH-GSSAPI]. Such pseudo-mechanisms are being proposed separately. [NOTE: Indirect reference to CCM...]

However, it does not, at this time, seem feasible to use SPNEGO with such pseudo-mechanisms for negotiating the use of channel bindings.

4.2 SASL and channel bindings

SASL does not provide for the use of channel bindings during initialization of SASL contexts.

SASL applications MAY define their own exchange of integrity-protected channel bindings using established SASL integrity layers.

Alternatively, SASL applications MAY use the GSS-* SASL mechanisms (which correspond to GSS-API mechanisms) to ensure the use of channel bindings through the GSS-API's facilities; this approach may require more study and specification elsewhere.

5. Channel bindings for various secure layers

Not every secure session protocol or interface provides for secure channels, and not every secure session protocol provides data suitable for use as channel bindings.

5.1 Bindings to SSHv2 channels

SSHv2 provides both, a secure channel and material (the SSHv2 "session ID") that is suitable for use as channel bindings.

Thus it is RECOMMENDED that the SSHv2 "session ID" be used as the channel bindings for SSHv2.

5.2 Bindings to TLS channels

TLS provides both, a secure channel and material (the TLS "finished" messages), that is suitable for use as channel bindings.

Thus it is RECOMMENDED that the concatenation of the client's and server's "finished" messages, in that order, be used as the channel bindings for TLS.

Note that the TLS "session ID," in spite of being named similarly to the SSHv2 session ID, is not suitable for use as channel bindings because it is assigned by the server, so a MITM could assign the same session ID on the client side as it gets from the server.

5.3 Bindings to IPsec

IPsec does not provide for secure channels by itself, as it protects individual packets. Further, the IPsec SAs used to protect the packets for some channel (e.g., a TCP connection) need not be related in any way that allows for construction of channel bindings.

There is a set of IPsec parameters that may be kept constant for all IP packets for a given channel (e.g., a TCP connection):

- the peers' authenticated IPsec IDs
- the SA types (e.g., transport mode ESP)
- the privacy and integrity protection algorithms used

[QUESTION: Should IPsec traffic selectors, that is, the protocol (TCP, UDP, SCTP) and port numbers used for the channel be included?]

Provided interfaces for binding a channel to these IPsec parameters it is possible to construct a channel secured by IPsec.

The channel bindings for such a channel, then, are the values of those IPsec parameters to which the channel is bound.

Requirements for such interfaces to IPsec are specified in [IPSP-APIREQ].

5.3.1 Interfaces for creating IPsec channels

In order to build an IPsec channel some additional application programming interfaces are needed to:

- indicate that an as yet unconnected channel is to be bound to IPsec IDs and
- explicitly specify one, the other or both of those IDs
- implicitly specify one, the other or both of those IDs (e.g., the ID corresponding to the current application program instance)
- indirectly specify one, the other or both of those IDs (e.g., through IP addresses or hostnames)

and/or

- discover the IPsec IDs to which a channel is bound

For connection-less datagram transports the IDs to be used need to be specified/discovered on a per-datagram basis.

See [IPSP-APIREQ].

5.4 Bindings to other types of channels

Channel bindings for other secure session protocols are not specified here.

6. Benefits of channel bindings to secure channels

The use of channel bindings to delegate session cryptographic protection include:

- Performance improvements by avoiding double protection of application data in cases where IPsec is in use and applications provide their own secure channels.
- Performance improvements by leveraging hardware-accelerated IPsec.
- Performance improvements by allowing RDDP hardware offloading to be integrated with IPsec hardware acceleration.

  Where protocols layered above RDDP use privacy protection RDDP offload cannot be done, thus by using channel bindings to IPsec the privacy protection is moved to IPsec, which is layered below RDDP, so RDDP can address application protocol data that's in cleartext relative to the RDDP headers.
- Latency improvements for applications that multiplex multiple users onto a single channel, such as NFS w/ RPCSEC_GSS.

7. Security Considerations

When delegating session protection from one layer to another, one will almost certainly be making some session security trade-offs, such as using weaker cipher modes in one layer than might be used in the other. Implementors and administrators SHOULD understand these trade-offs.

Channel bindings cannot and MUST NOT be used without mutual authentication (of client/user/initiator and server/user/acceptor).

Anonymous secure channels SHOULD NOT be used without authentication and corresponding use of their channel bindings at higher network layers.

The security of channel bindings depends on the security of the channels, the construction of the bindings and the security of the authentication and integrity protection used to exchange channel bindings.

8. References

8.1 Normative

[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.

8.2 Informative

[RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2025] Adams, C., "The Simple Public-Key GSS-API Mechanism (SPKM)", RFC 2025, October 1996.

[RFC2203] Eisler, M., Chiu, A. and L. Ling, "RPCSEC_GSS Protocol Specification", RFC 2203, September 1997.

[RFC2478] Baize, E. and D. Pinkas, "The Simple and Protected GSS-API Negotiation Mechanism", RFC 2478, December 1998.

[RFC2623] Eisler, M., "NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5", RFC 2623, June 1999.

[RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M. and D. Noveck, "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003.

Author's Address

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US

EMail: Nicolas.Williams@sun.com

Appendix A. Acknowledgments

The author would like to thank Mike Eisler for his work on the Channel Conjunction Mechanism I-D and for bringing the problem to a head, Sam Hartman for pointing out that channel bindings provide a general solution to the channel binding problem, Jeff Altman for his suggestion of using the TLS finished messages as the TLS channel bindings, Bill Sommerfeld, for his help in developing channel bindings for IPsec, and Radia Perlman for her most helpful comments.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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