SASL And Channel Binding
draft-ietf-sasl-channel-bindings-03.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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 October 23, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document specifies the semantics of channel binding for the Simple Authentication and Security Layers (SASL) framework, mechanisms and applications. This includes negotiation of channel binding, and negotiation of channel binding types.

Table of Contents

1. Introduction ................. 3
1.1. Conventions used in this document ........ 3
2. Channel Binding Semantics and Negotiation for SASL .... 4
2.1. Channel Binding Negotiation .............. 4
2.2. Channel Binding Type Negotiation .......... 5
3. IANA Considerations ............... 7
4. Security Considerations .............. 8
5. References ..................... 9
5.1. Normative References ............... 9
5.2. Informative References ............... 9
Author’s Address .................. 10
1. Introduction

The introduction of the Salted Challenge Response (SCRAM) SASL mechanism [I-D.newman-auth-scram] and GS2 family of SASL mechanisms [I-D.ietf-sasl-gs2] requires that we define the semantics of channel binding [RFC5056] in the context of SASL [RFC4422]. Without such a definition we risk ending up with mechanism-specific channel binding code in applications.

In SASL channel bindings are all-or-nothing, and the use or non-use of channel binding is negotiated via mechanism negotiation, with downgrade protection built into mechanisms that support channel binding. See Section 2.

1.1. 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. Channel Binding Semantics and Negotiation for SASL

In SASL [RFC4422] channel binding [RFC5056] is all-or-nothing. If channel binding is used and channel binding fails then SASL authentication MUST also fail. This means that applications either must know a priori whether to use channel binding, or the must negotiate whether to use channel binding. To improve interoperability we therefore provide a method for negotiating the use of channel binding.

Secure channels may also export multiple types of channel bindings, such as "unique" channel bindings and "end-point" channel bindings [RFC5056]. Since channel binding is a fairly recent addition to protocols like TLS [RFC5246], and since the set of channel binding types is open, it is not necessarily the case that both ends of a secure end-to-end channel support the same set of channel binding types. In order to interoperate it is necessary that both ends agree on a channel binding type to use. Also, some SASL mechanisms, such as YAP [I-D.zeilenga-sasl-yap], may work only with "unique" channel bindings, but "end-point" channel bindings are also quite desirable as they have better interoperability characteristics when server-side TLS proxies ("concentrators") are used (see [tls-server-end-point]). Therefore we also provide for negotiation of channel binding types.

2.1. Channel Binding Negotiation

Negotiation of whether to use channel binding is achieved by overloading the SASL mechanism negotiation. Each SASL mechanism name MUST imply whether the channel binding is supported. This means that mechanisms which support optional use of channel binding MUST have two SASL mechanism names. See Section 3 for more information.

The server application MUST advertise SASL mechanism names that correspond to whether the secure channel, if any, and the application support channel binding; conversely, the server MUST NOT advertise SASL mechanism names which indicate the opposite. For example, if the server has support for SCRAM [I-D.newman-auth-scram], there is a TLS channel, and the server application supports channel binding, then the server application must advertise the SCRAM mechanism name that indicates support for channel binding, and it must not advertise the SCRAM mechanism name that indicates the opposite.

The client MUST NOT use channel binding if it lists the server’s mechanisms and does not find a suitable mechanism that supports channel binding in that list.

To prevent downgrade attacks each mechanism that supports channel binding MUST provide downgrade attack detection. To do this the
client application MUST provide the name of the selected mechanism, or the server’s entire mechanism list, as an input to the mechanism prior to producing the mechanism’s first authentication message. The mechanism MUST securely indicate to the server whether the client a) chose to use channel binding, b) would have chosen to use channel binding if the server had supported it, c) cannot do channel binding. In the case of (c) the server MUST fail authentication if the server does actually support channel binding, as that would be an indication of a downgrade attack.

2.2. Channel Binding Type Negotiation

The negotiation of what channel binding type to use is also achieved by overloading the SASL mechanism negotiation. In this case we use pseudo-mechanism names to indicate what channel binding types are available on the server side. See Section 3 for more information on these pseudo-mechanism names.

The server application SHOULD advertise SASL pseudo-mechanism names corresponding to the channel binding types that are available on the server end of the secure channel to be bound into SASL authentication.

The IANA registrations of SASL mechanisms that support channel binding MUST indicate whether the mechanism requires "unique" channel binding types. Given this information it is possible for the client to select a channel binding type that is available locally and on the server side. If the server did not advertise any channel binding types but did advertise mechanisms that support channel binding, then the client SHOULD assume that all locally available channel binding types are also available on the server side.

Whether the application, SASL framework, or SASL mechanism decides which channel binding type to use is an implementation detail. An implementation could have the application provide the server’s mechanism list to a SASL framework which then decides which channel binding type to use, or perhaps the framework will then pass that list to the chosen mechanism which will then in turn decide. It’s also possible to let the application decide. All three implementation designs are made possible by including an indication, in a mechanism’s SASL name registration, of whether the mechanism requires the use of unique channel binding types.

The channel binding type negotiation is protected only by protecting the SASL mechanism name negotiation, which, if done over the secure channel is protected by that channel and, therefore, by channel binding.
Note that while the channel binding negotiation is done through concrete mechanism name negotiation, the channel binding type negotiation is done through pseudo-mechanism names. Both could have been done via pseudo-mechanism names, however, in the interest of making the simplest cases simple and obvious we used concrete mechanism names for the channel binding negotiation.
3. IANA Considerations

This document changes the procedures for registration of SASL mechanism names in the IANA SASL mechanism name registry.

Henceforth any SASL mechanism registration MUST include one or two mechanism names and an indication of which name indicates server support for channel binding. Also REQUIRED of all new SASL mechanism registrations is a note indicating whether the mechanism requires the use of channel binding, and another note indicating whether the mechanism requires the use of unique channel binding types.

Additionally, the IANA is directed to review and allow SASL pseudo-mechanism name registrations corresponding to channel binding types in the IANA channel binding type registry. The following pseudo-mechanism names are to be added to the registry immediately:

- ‘CB-tls-srv-endpoint’ (corresponding to the ‘tls-server-end-point’ [tls-server-end-point] channel binding type);
- ‘CB-tls-unique’ (corresponding to the ‘tls-unique’ [tls-unique] channel binding type).
4. Security Considerations

For general security considerations relating to channel bindings see [RFC5056]. For general security considerations relating to SASL see [RFC4422].

This document specifies how channel binding fits into SASL and, specifically, the semantics of channel binding for SASL and how the use of channel binding and which channel binding type to use are negotiated. The negotiation of channel binding is subject to downgrade attacks by active attackers, therefore we include a requirement that SASL mechanisms provide protection against downgrade attacks. Protection against downgrade attacks requires that the application provide certain information to the SASL mechanism. See Section 2.

The negotiation of channel binding type is protected by channel binding, assuming that SASL mechanism negotiation is done with integrity protection from the secure channel.
5. References

5.1. Normative References


5.2. Informative References


Author’s Address

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

Email: Nicolas.Williams@sun.com