XEP-0440: SASL Channel-Binding Type Capability

Florian Schmaus
mailto:flo@geekplace.eu
xmpp:flo@geekplace.eu

2020-08-04
Version 0.2.0

Status	Type	Short Name
Experimental	Standards Track	sasl-cb-types

This specification allows servers to annouce their supported SASL channel-binding types to clients.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF’s Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1 Introduction

SASL channel-binding is a technique to increase the security of connections (RFC 5056 [1]). Unfortunately, the SASL profile specified in RFC 6120 [2] lacks a method for the server to announce its supported channel-binding types. This hinders the adoption of channel-binding, especially since the error protocol to execute after a client requested a channel-binding type unsupported by the server is basically unspecified.
The extension defined herein fills the gap left by RFC 6120 [3], by allowing the server the announce its supported channel-binding types.

2 Announcing the SASL Channel-Binding Type Capability

This protocol consists of a single optional extension element named 'sasl-channel-binding' qualified by the 'urn:xmpp:sasl-cb:0' namespace. The 'sasl-channel-binding' element MUST contain one or more 'channel-binding' elements, of which each MUST have an attribute with the name 'type'. The value of the 'type' attribute SHOULD be the "Channel-binding unique prefix" of a channel-binding type which was registered with the IANA Channel-Binding Types Registry [4].
A server declares that it supports particular channel-binding types by listing the supported types via the 'sasl-channel-binding' element defined herein. The 'sasl-channel-binding' element could appear as child element to the SASL <mechanisms/> stream-feature element, qualified by the 'urn:ietf:params:xml:ns:xmpp-sasl' namespace, as specified in RFC 6120 [5]. Another potential appearance of <sasl-channel-binding> is as child element of the <mechanisms/> stream-feature element as specified in the Extensible SASL Profile (XEP-0388) [6].

Listing 1: Example <mechanisms/> stream feature with SASL Channel-Binding Type Capability.

```
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>EXTERNAL</mechanism>
    <mechanism>SCRAM-SHA-1-PLUS</mechanism>
    <mechanism>PLAIN</mechanism>
    <sasl-channel-binding xmlns='urn:xmpp:sasl-cb:0'>
      <channel-binding type='tls-server-end-point'/>
      <channel-binding type='tls-exporter'/>
    </sasl-channel-binding>
  </mechanisms>
```

[1] RFC 5056: On the Use of Channel Bindings to Secure Channels <http://tools.ietf.org/html/rfc5056>.
[2] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[3] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[4] IANA Channel-Binding Types Registry <https://www.iana.org/assignments/channel-binding-types/channel-binding-types.xhtml>.
[5] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[6] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.

```
</stream:features>
```

3 Interaction with SASL mechanisms

Some channel-binding enabled SASL mechanisms reflect the server's presumed channel-binding abilities back to the server. This prevents SASL-mechanism stripping attacks, where a Man in the Middle (MITM) removes certain SASL mechanisms in an attempt to downgrade the mechanism choosen for authentication to a non-channel-binding enabled one. An example of a SASL mechanism family with this feature is RFC 5802 [7]. This standard specifies the gs2-cbind-flag. The flag has a tristate value of "I don't support channel-binding" (n), "I think you do not support channel-binding, but I do" (y), or, "Let us use channel-binding type X" (p). Clients using the information provided via <sasl-channel-binding/> MAY want to indicate to the server that they do not support channel-binding (even if they do) if no mutual supported channel-binding type was found. The only alternative is, that the client signals the server that he believes that the server does not support channel binding. But this may cause the server to terminate the connection, because it indicates a potential ongoing SASL-mechanism stripping attack.

4 Security Considerations

If a client signals to the server that he does not support channel binding, because it found no mutual supported channel-binding types, another MITM attack vector is introduced. An active attacker could replace the <sasl-channel-binding;> list with channel bindings unlikely (or impossible) to be supported by the client. If the client is configured to use non-channel-binding SASL mechanisms as a fallback, this could be used to downgrade the connection security. Note that this attack is a different one than the SASL-mechanism stripping one: Here the attacker tempers with the announced channel-binding types, i.e., the values within <sasl-channel-binding;>

Depending on the application's security policy, clients may refrain from falling back to non-channel-binding SASL mechanisms if no mutual supported channel-binding type is available. Alternatively, they may try channel-binding with a supported type nevertheless. To mitigate the attack describe above, clients could "pin" the announced channel bindings types by a service. In that case, implementations may want to allow the set of pinned channel-binding types to be extended to stronger ones.

As further mitigation, it is RECOMMENDED to implement the channel-binding type tls-server-end-point (RFC 5929 [8]) to increase the probability of a mutual supported channel-binding type.

[7] RFC 5802: Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms <http://tools.ietf.org/html/rfc5802>.

[8] RFC 5929: Channel Bindings for TLS <http://tools.ietf.org/html/rfc5929>.

5 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [9].

6 XMPP Registrar Considerations

This document requires no interaction with the XMPP registrar.

7 XML Schema

TODO: Add if the XEP is scheduled for the state after 'experimental'.

8 Acknowledgements

Thanks to Sam Whited for the discussion about the underlying issue and incentivizing me to come up with this extension. Further thanks goes to Ruslan N. Marchenko for pointing out the possible MITM attack vector. Last but not least, Dave Cridland provided valuable feedback.

[9] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.