Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 6, 2016

R. Van Rein
ARPA2.net

September 3, 2015

SASL Security for Remote Framebuffers
draft-vanrein-kitten-rfbsasl-00

Abstract

The Remote Framebuffer Protocol is widely used to provide remote access to desktops. This specification defines security levels that are in line with these usage patterns.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on March 6, 2016.

Table of Contents

1. Introduction

The Remote Framebuffer Protocol (RFB) [RFC6143] is popular, but it only defines weak security mechanisms. This is particularly problematic because the protocol is often used for remote administration, including remote support to users who lack the knowledge to verify the actions taken over the protocol.

The existing protection of RFB is founded on passwords, that may be session-specific, but communicated over unprotected media. Once again, the users that open their desktops for remote administration are not always in a position to share such passwords securely.

This specification introduces a new Security Type for RFB, by introducing SASL into the authentication phase and, when available, employing its security layer for encryption after bootstrapping the connection. SASL enables the use of strong encryption mechanisms. Encryption is termed a "security layer" in SASL.

A SASL mechanism worth mentioning is GSS-API, which introduces two commonly used families of security mechanisms. The first is Kerberos5 [RFC4121], the other is EAP [RFC7055], which in turn ties in with common authentication infrastructures such as RADIUS [RFC3579] and Tunneled TLS [RFC5281]. These mechanisms are supportive of centralised management of access rights to RFB sessions.

For use with Kerberos, a service ticket SHOULD use the service name "rfb", so a service ticket could have a principal name like "rfb/laptop.example.com@EXAMPLE.COM". Beyond this service name prefix, this principal name example is not prescriptive; so, this specification does not exclude the use of additional descriptive levels with / or @ in any Kerberos-compliant manner to name independent sessions and/or user names.

2. SASL Security Type Requirements

Clients and servers may independently be configured to require either or both of encryption and authentication of their remote; SASL mechanisms exist that support mutual authentication and encryption, so mutual authentication is possible. In addition, it is worth noting that authentication mechanisms can usually be extended to incorporate Diffie-Hellman for encryption. This applies to both the modular-exponential and elliptic-curve forms. Since such work falls in the scope of the SASL specifications, we shall not define such modifications in this specification, but future extensions could be helpful for the RFB application of SASL.

Among the SASL mechanisms is the EXTERNAL mechanism [Appendix A of [RFC4422]] that can be used to refer to a wrapping protocol, such as the TLS [RFC5246] protocol that provides both encryption and mutual authentication.

3. SASL Security Type Definition

Immediately preceding the SASL-specific handshake, the client and server exchange a Security Handshake [Section 7.1.2 of [RFC6143]] in which the server offers security-types that include TBD, and the client selects security-type TBD.

In response to the client selection of security-type TBD, the server sends a list of SASL mechanisms that it supports. The mechanisms are listed by their sasl-mech name [Section 3.1 of [RFC4422]] with a space character %x20 separating the alternatives. The format used for this information is:

No. of bytes	Type [Value]	Description
2	U16	sasl-mech-length
sasl-mech-length	U8 array	sasl-mech-string

The client now selects a mechanism from the space-separated list that the server offered, and sends the chosen sasl-mech name back to the server, using the same format, with the exception that only a single sasl-mech-string is sent and so no space characters %x20 occur in the sasl-mech-string.

The SASL-specific exchange is then initiated by the client, and messages are passed back and forth until the server casts a final decision [Section 3 of [RFC4422]]. The information is sent in its binary form, without a need to use a transport encoding such as

base64. It is however framed in the following format, in both directions:

No. of bytes	Type [Value]	Description
2	U16	sasl-msg-length
sasl-msg-length	U8 array	sasl-message

After the SASL exchange finishes, the server sends an RFB-style SecurityResult Handshake [Section 7.1.3 of [RFC6143]] and continues as it would for other security mechanisms. The server MUST NOT send conflicting results in the final SASL message and the SecurityResult messages. An aborted SASL exchange MUST be treated equivalently to a failed authentication attempt.

Starting with the Initialization Messages [Section 7.4 of [RFC6143]], the protocol MUST be sent through the security layer defined for the SASL mechanism. Only the EXTERNAL method is exempt from this requirement, under the assumption that it is run inside a layer that already encrypts the message flow. Applications SHOULD assure that this is indeed the case.

Some forms of encryption require framing when they are transported over a byte sequence abstraction such as offered by TCP; this is dealt with in the SASL specification [Section 3.7 of [RFC4422]] with a length prefix to the buffers being transmitted. This means that no further framing is required of the RFB protocol, but implementations may interpret SASL framing when encryption is employed.

An RFB implementation using the SASL security-type MUST provide each RFB message separately to the SASL layer for mapping it to a SASL message; when receiving, an RFB implementation MAY require that each SASL message represents precisely one RFB message. On the wire, SASL messages are transmitted in its binary octet form, without further transport encodings such as base64.

4. IANA Considerations

This specification defines a security type named SASL, registered by IANA in the registry for Remote Framebuffer Security Types with numeric identifier TBD.

5. Security Considerations

Not all parts of the protocol described here are protected. The unprotected parts are subject to various forms of attack, including downgrade attacks and denial-of-service attacks. These risks apply to the RFB protocol version, the entire SASL exchange and the RFB SecurityResult.

6. References

6.1. Normative References

[RFC4422] Melnikov, A., Ed. and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, DOI 10.17487/RFC4422, June 2006, <http://www.rfc-editor.org/info/rfc4422>.

[RFC6143] Richardson, T. and J. Levine, "The Remote Framebuffer Protocol", RFC 6143, DOI 10.17487/RFC6143, March 2011, <http://www.rfc-editor.org/info/rfc6143>.

6.2. Informative References

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, DOI 10.17487/RFC3579, September 2003, <http://www.rfc-editor.org/info/rfc3579>.

[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, DOI 10.17487/RFC4121, July 2005, <http://www.rfc-editor.org/info/rfc4121>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <http://www.rfc-editor.org/info/rfc5281>.

[RFC7055] Hartman, S., Ed. and J. Howlett, "A GSS-API Mechanism for the Extensible Authentication Protocol", RFC 7055, DOI 10.17487/RFC7055, December 2013, <http://www.rfc-editor.org/info/rfc7055>.

Author's Address

Rick van Rein
ARPA2.net
Haarlebrink 5
Enschede, Overijssel 7544 WP
The Netherlands

Email: rick@openfortress.nl