Kerberos Working Group
Category: Informational
Document: draft-raeburn-krb-gssapi-krb5-3des-01.txt

K. Raeburn
MIT

November 24, 2000

Triple-DES Support for the Kerberos 5 GSSAPI Mechanism

Status of this Memo

1. Abstract

The GSSAPI Kerberos 5 mechanism definition [GSSAPI-KRB5] specifically enumerates encryption and checksum types, independently of how such schemes may be used in Kerberos. In the long run, a new Kerberos-based mechanism, which does not require separately enumerating for the GSSAPI mechanism each of the various encryption types defined by Kerberos, is probably a better approach. Various people have expressed interest in designing one, but the work has not yet been completed.

The MIT Kerberos 5 release version 1.2 includes support for triple-DES with key derivation [KrbRev]. Recent work by the EFF [EFF] has demonstrated the vulnerability of single-DES mechanisms to brute-force attacks by sufficiently motivated and well-funded parties. So, in the interest of providing increased security in the near term, MIT is adding support for triple-DES to the existing mechanism implementation we ship, as an interim measure.

2. New Algorithm Identifiers

One new sealing algorithm is defined, for use in Wrap tokens.

name	octet values
DES3-KD	02 00

This algorithm uses triple-DES with key derivation, with a usage value KG_USAGE_SEAL. (Unlike the EncryptedData definition in [KrbRev], no integrity protection is needed, so this is "raw" triple-DES, with no checksum attached to the encrypted data.) Padding is still to 8-byte multiples, and the IV for encrypting application data is zero.

One new signing algorithm is defined, for use in MIC, Wrap, and Delete tokens.

name	octet values
HMAC SHA1 DES3-KD	04 00

This algorithm generates an HMAC using SHA-1 and a derived DES3 key with usage KG_USAGE_SIGN, as described in [KrbRev].

[N.B.: The current [KrbRev] description refers to expired I-Ds from Marc Horowitz. The text in [KrbRev] may be inadequate to produce an interoperable implementation.]

The checksum size for this algorithm is 20 octets. See section 4.3 below for the use of checksum lengths of other than eight bytes.

3. Key Derivation

For purposes of key derivation, we add three new usage values to the list defined in [KrbRev]; one for signing messages, one for sealing messages, and one for encrypting sequence numbers:

name	value
KG_USAGE_SEAL	22
KG_USAGE_SIGN	23
KG_USAGE_SEQ	24

4. Adjustments to Previous Definitions

4.1. Quality of Protection

The GSSAPI specification [GSSAPI] says that a zero QOP value indicates the "default". The original specification for the Kerberos 5 mechanism says that a zero QOP value (or a QOP value with the appropriate bits clear) means DES encryption.

Rather than forcing the use of plain DES when the application doesn't use mechanism-specific QOP values, we redefine the explicit DES QOP value as a non-zero value, and define a triple-DES value as well. Then a zero value continues to imply the default, which would be triple-DES protection when given a triple-DES session key.

Our values are:

name	value	meaning
GSS_KRB5_INTEG_C_QOP_HMAC_SHA1	0x0004	SHA-1 HMAC, using key derivation
GSS_KRB5_CONF_C_QOP_DES	0x0100	plain DES encryption
GSS_KRB5_CONF_C_QOP_DES3_KD	0x0200	triple-DES with key derivation

Rather than attempt to specify a generic mechanism for deriving a key of one type given a key of another type, and evaluate the security implications of using a short key to generate a longer key to satisfy the requested quality of protection, our implementation will simply

return an error if the nonzero QOP value specified does not correspond to the session key type.

4.2. MIC Sequence Number Encryption

The sequence numbers are encrypted in the context key (as defined in [GSSAPI-KRB5] -- this will be either the Kerberos session key or asubkey provided by the context initiator), using whatever encryption system is designated by the type of that context key. The IV is formed from the first N bytes of the SGN_CKSUM field, where N is the number of bytes needed for the IV. (With all algorithms described here and in [GSSAPI-KRB5], the checksum is at least as large as the IV.)

4.3. Message Layout

Both MIC and Wrap tokens, as defined in [GSSAPI-KRB5], contain an checksum field SGN_CKSUM. In [GSSAPI-KRB5], this field was specified as being 8 bytes long. We now change this size to be "defined by the checksum algorithm", and retroactively amend the descriptions of all the checksum algorithms described in [GSSAPI-KRB5] to explicitly specify 8-byte output. Application data continues to immediately follow the checksum field in the Wrap token.

The revised message descriptions are thus:

MIC token:

Byte #	Name	Description
0..1	TOK_ID	Identification field.
2..3	SGN_ALG	Integrity algorithm indicator.
4..7	Filler	Contains ff ff ff ff
8..15	SND_SEQ	Sequence number field.
16..s+15	SGN_CKSUM	Checksum of "to-be-signed data", calculated according to algorithm specified in SGN_ALG field.

Wrap token:

Byte #	Name	Description
0..1	TOK_ID	Identification field. Tokens emitted by GSS_Wrap() contain the hex value 02 01 in this field.
2..3	SGN_ALG	Checksum algorithm indicator.
4..5	SEAL_ALG	Sealing algorithm indicator.
6..7	Filler	Contains ff ff
8..15	SND_SEQ	Encrypted sequence number field.
16..s+15	SGN_CKSUM	Checksum of plaintext padded data, calculated according to algorithm specified in SGN_ALG field.
s+16..last	Data	encrypted or plaintext padded data

Where "s" indicates the size of the checksum.

As indicated above in section 2, we define the HMAC SHA1 DES3-KD checksum algorithm to produce a 20-byte output, so encrypted data begins at byte 36.

5. Backwards Compatibility Considerations

The context initiator should request of the KDC credentials using session-key cryptosystem types supported by that implementation; if the only types returned by the KDC are not supported by the mechanism implementation, it should indicate a failure. This may seem obvious, but early implementations of both Kerberos and the GSSAPI Kerberos mechanism supported only DES keys, so the cryptosystem compatibility question was easy to overlook.

Under the current mechanism, no negotiation of algorithm types occurs, so server-side (acceptor) implementations cannot request that clients not use algorithm types not understood by the server. However, administration of the server's Kerberos data (e.g., the service key) has to be done in communication with the KDC, and it is from the KDC that the client will request credentials. The KDC could therefore be tasked with limiting session keys for a given service to types actually supported by the Kerberos and GSSAPI software on the server.

This does have a drawback for cases where a service principal name is used both for GSSAPI-based and non-GSSAPI-based communication (most notably the "host" service key), if the GSSAPI implementation does not understand triple-DES but the Kerberos implementation does. It means that triple-DES session keys cannot be issued for that service

principal, which keeps the protection of non-GSSAPI services weaker than necessary.

It would also be possible to have clients attempt to get single-DES session keys before trying to get triple-DES session keys, and have the KDC refuse to issue the single-DES keys only for the most critical of services, for which single-DES protection is considered inadequate. However, that would eliminate the possibility of connecting with the more secure cryptosystem to any service that can be accessed with the weaker cryptosystem.

For MIT's 1.2 release, we chose to go with the former approach, putting the burden on the KDC administration and gaining the best protection possible for GSSAPI services, possibly at the cost of weaker protection of non-GSSAPI Kerberos services running earlier versions of the software.

6. Security Considerations

Various tradeoffs arise regarding the mixing of new and old software, or GSSAPI-based and non-GSSAPI Kerberos authentication. They are discussed in section 5.

7. References

[EFF] Electronic Frontier Foundation, "Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design", O'Reilly & Associates, Inc., May, 1998.

[GSSAPI] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January, 2000.

[GSSAPI-KRB5] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June, 1996.

[KrbRev] Neuman, C., Kohl, J., Ts'o, T., "The Kerberos Network Authentication Service (V5)", draft-ietf-cat-kerberos-revisions-06.txt, July 4, 2000.

8. Author's Address

Kenneth Raeburn Massachusetts Institute of Technology 77 Massachusetts Avenue Cambridge, MA 02139

9. Full Copyright Statement

10. Document Change History

>From -00 to -01:

Converted master to GNU troff and tbl, rewriting tables in the process.

Specify informational category only. Modify some text to emphasize that this document intends to describe MIT's extensions.

Point out that while EncryptedData for 3des-kd includes a checksum, DES3-KD GSS encryption does not.

Shorten backwards-compatibility descriptions a little.

Submit to Kerberos working group rather than CAT.