Network Working Group
Internet-Draft
Obsoletes: 4757 (if approved)
Updates: 3961 (if approved)
Intended status: Best Current Practice
Expires: December 6, 2017

B. Kaduk
Akamai
M. Short
Microsoft Corporation

June 4, 2017

Deprecate 3DES and RC4 in Kerberos
draft-ietf-curdle-des-des-des-die-die-die-02

Abstract

The 3DES and RC4 encryption types are steadily weakening in cryptographic strength, and the deprecation process should be begun for their use in Kerberos. Accordingly, RFC 4757 is moved to Obsolete status, as none of the encryption types it specifies should be used, and RFC 3961 is updated to note the deprecation of the triple-DES encryption types.

Status of This Memo

This Internet-Draft will expire on December 6, 2017.

Copyright Notice

Table of Contents

1. Introduction

The 3DES and RC4 encryption types are steadily weakening in cryptographic strength, and the deprecation process should be begun for their use in Kerberos. Accordingly, RFC 4757 is moved to Obsolete status, as none of the encryption types it specifies should be used, and RFC 3961 is updated to note the deprecation of the triple-DES encryption types.

2. Requirements Notation

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].

3. Affected Specifications

The RC4 Kerberos encryption types are specified in [RFC4757], which is moved to historic.

The des3-cbc-sha1-kd encryption type is specified in [RFC3961]. Additional 3DES encryption types are in use with no formal specification, in particular des3-cbc-md5 and des3-cbc-sha1. These unspecified encryption types are also deprecated by this document.

4. Affected Encryption Types

The following encryption types are deprecated. The numbers are the official identifiers; the names are only for convenience.

enctype number	enctype convenience name
5	des3-cbc-md5
7	des3-cbc-sha1
16	des3-cbc-sha1-kd
23	rc4-hmac

5. RC4 Weakness

RC4's weakness as a TLS cipher due to statistical biases in the keystream has been well-publicized [RFC7465], and these statistical biases cause concern for any consumer of the RC4 cipher. However, the RC4 Kerberos enctypes have additional flaws which reduce the security of applications using them, including the weakness of the password hashing algorithm, the reuse of key material across protocols, and the lack of a salt when hashing the password.

5.1. Statistical Biases

The RC4 stream cipher is known to have statistical biases in its output, which have led to practical attacks against protocols using RC4, such as TLS ([RFC7465]). These attacks seem to rely on repeated encryptions of thousands of copies of the same plaintext; whereas it is easy for malicious javascript in a website to cause such traffic, it is unclear that there is an easy way to induce a kerberized application to generate such repeated encryptions. The statistical biases are most pronounced for earlier bits in the output stream, which is somewhat mitigated by the use of a confounder in kerberos messages -- the first 64 bits of plaintext are a random confounder, and are thus of no use to an attacker who can retrieve them.

Nonetheless, the statistical biases in the RC4 keystream extend well past 64 bits, and provide potential attack surface to an attacker.

Continuing to use a known weak algorithm is inviting further development of attacks.

5.2. Password Hash

Kerberos long-term keys can either be random (as might be used in a service's keytab) or derived from a password (usable for individual users to authenticate to a system). The specification for a Kerberos encryption type must include a "string2key" algorithm for generating a raw crypto key from a string (i.e., password). Modern encryption types, such as those using the AES and Camellia block ciphers, use a string2key function based on the PBKDF2 algorithm, which involves many iterations of a cryptographic hash function, designed to increase the computational effort required to perform a brute-force password-guessing attack. There is an additional option to specify an increased iteration count for a given principal, providing some modicum of adaptability for increases in computing power.

It is also best practice, when deriving cryptographic secrets from user passwords, to include a value which is unique to both the user and the realm of authentication as input to the hash function; this user-specific input is known as a "salt". The default salt for Kerberos principals includes both the name of the principal and the name of the realm, in accordance with these best practices. However, the RC4 encryption types ignore the salt input to the string2key function, which is a single iteration of the MD4 HMAC function applied to the UTF-16 encoded password, with no salt at all. The MD4 hash function is very old, and is considered to be weak and unsuitable for new cryptographic applications at this time. [RFC6150]

The omission of a salt input to the hash is contrary to cryptographic best practices, and allows an attacker to construct a "rainbow table" of password hashes, which are applicable to all principals in all Kerberos realms. Given the prevalance of poor-quality user-selected password, it is likely that a rainbow table derived from a database of common passwords would be able to compromise a sizable number of Kerberos principals in any realm using RC4 encryption types for password-derived keys.

5.3. Cross-Protocol Key Reuse

The selection of unsalted MD4 as the Kerberos string2key function was deliberate, since it allowed systems to be converted in-place from the old NTLM logon protocol [MS-NLMP] to use Kerberos.

Unfortunately, there still exist systems using NTLM for authentication to applications, which can result in application

servers possessing the NT password hash of user passwords. Because the RC4 string2key was chosen to be compatible with the NTLM scheme, these application servers also possess the long-term Kerberos key for those users even though the password is unknown. The cross-protocol use of the long-term key/password hash was convenient for migrating to Kerberos, but now provides a vulnerability in Kerberos as NTLM continues to be used.

5.4. Interoperability Concerns

The RC4 Kerberos encryption type remains in use in many environments because of interoperability requirements -- in those sites, RC4 is the strongest enctype which allows two parties to use Kerberos to communicate. In particular, the Kerberos implementions included with Windows XP and Windows Server 2003 support only single-DES and RC4. Since single-DES is deprecated ([RFC6649]), machines running those operating systems must use RC4.

Similarly, there are cross-realm deployments where the cross-realm key was initially established when one peer only supported RC4, or where machines only supporting RC4 will need to obtain a cross-realm Ticket-Granting Ticket. It can be difficult to inventory all clients in a Kerberos realm and know what implementations will be used by those client principals; this leads to concerns that disabling RC4 will cause breakage on machines that are unknown to the realm administrators.

However, both Windows XP and Windows Server 2003 are already out of their official support periods. It is now believed that all machines that might be broken by disabling RC4 are unsupported, and concerns about breaking them will be reduced. That should facilitate the removal of RC4 from common use.

6. 3DES Weakness

The flaws in triple-DES as used for Kerberos are not quite as damning as those in RC4, but there is still ample justification for deprecating its use. As is the case for the RC4 enctypes, the string2key algorithm is weak. Additionally, the 3DES encryption types were never implemented in all Kerberos implementations, and the 64-bit block size may be problematic in some environments.

6.1. Password-based Keys

The n-fold-based string2key function used by the des3-cbc-sha1-kd encryption type is an ad-hoc construction that should not be considered cryptographically sound. It is known to not provide effective mixing of the input bits, and is computationally easy to

evaluate. As such, it does not slow down brute-force attacks in the way that the computationally demanding PBKDF2 algorithm used by more modern encryption types does. The salt is used by des3-cbc-sha1-kd's string2key, in contrast to RC4, but a brute-force dictionary attack on common passwords may still be feasible.

6.2. Block Size

Because triple-DES is based on the single-DES primitive, just using additional key material and nested encryption, it inherits the 64-bit cipher block size from single-DES. As a result, an attacker who can collect approximately 2**32 blocks of ciphertext has a good chance of finding a cipher block collision (the "birthday attack"), which would potentially reveal a couple blocks of plaintext.

A cipher block collision would not necessarily cause the key itself to be leaked, so the plaintext revealed by such a collision would be limited. For some sites, that may be an acceptable risk, but it is still considered a weakness in the encryption type.

6.3. Interoperability

The triple-DES encryption types were implemented by MIT Kerberos early in its development (ca. 1999) and present in the 1.2 release, but encryption types 17 and 18 (AES) were implemented by 2003 and present in the 1.3 release. The Heimdal Kerberos implementation also provided a version of 3DES in 1999 (though the GSSAPI portions remained non-interoperable with MIT for some time after that), and gained support for AES in 2005 with its 0.7 release. Both Heimdal and MIT krb5 have supported the AES enctypes for some 12 years, and it is expected that deployments that support 3DES but not AES are quite rare.

The Kerberos implementation in Microsoft Windows does not currently and has never implemented the 3DES encryption type. Support for AES was introduced with Windows Vista and Windows Server 2008; older versions such as Windows XP and Windows Server 2003 only supported the RC4 encryption types.

The 3DES encryption type offers very slow encryption, especially compared to the performance of AES using the hardware accelleration available in modern CPUs. There are no areas where it offers advantages over other encryption types except in the rare case where AES is not available.

7. Recommendations

This document hereby removes the following RECOMMENDED types from [RFC4120]:

Encryption: DES3-CBC-SHA1-KD

Checksum: HMAC-SHA1-DES3-KD

Kerberos implementations and deployments SHOULD NOT implement or deploy the following triple-DES encryption types: DES3-CBC-MD5(5), DES3-CBC-SHA1(7), and DES3-CBC-SHA1-KD(16) (updates [RFC4120]).

Kerberos implementations and deployments SHOULD NOT implement or deploy the RC4 encryption type RC4-HMAC(23).

Kerberos implementations and deployments SHOULD NOT implement or deploy the following checksum types: RSA-MD5(7), RSA-MD5-DES3(9), HMAC-SHA1-DES3-KD(12), and HMAC-SHA1-DES3(13) (updates [RFC4120]).

Kerberos GSS mechanism implementations and deployments SHOULD NOT implement or deploy the following SGN_ALGs: HMAC MD5(1100) and HMAC SHA1 DES3 KD (updates [RFC4757]).

Kerberos GSS mechanism implementations and deployments SHOULD NOT implement or deploy the following SEAL_ALGs: RC4(1000) and DES3KD(0400).

This document recommends the reclassification of [RFC4757] as Historic.

8. Security Considerations

This document is entirely about security considerations, namely that the use of the 3DES and RC4 Kerberos encryption types is not secure, and they should not be used.

9. IANA Considerations

IANA is requested to update the registry of Kerberos Encryption Type Numbers [IANA-KRB] to note that encryption types 1, 2, 3, and 24 are deprecated, with RFC 6649 ([RFC6649]) as the reference, and that encryption types 5, 7, 16, and 23 are deprecated, with this document as the reference.

Similarly, IANA Is requested to update the registry of Kerberos Checksum Type Numbers [IANA-KRB] to note that checksum types 1, 2, 3, 4, 5, 6, and 8 are deprecated, with RFC 6649 as the reference, and

that checksum types 7, 12, and 13 are deprecated, with this document as the reference.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, DOI 10.17487/RFC3961, February 2005, <http://www.rfc-editor.org/info/rfc3961>.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, DOI 10.17487/RFC4120, July 2005, <http://www.rfc-editor.org/info/rfc4120>.

[RFC6150] Turner, S. and L. Chen, "MD4 to Historic Status", RFC 6150, DOI 10.17487/RFC6150, March 2011, <http://www.rfc-editor.org/info/rfc6150>.

10.2. Informative References

[IANA-KRB]
Internet Assigned Numbers Authority, "IANA Kerberos Parameters Registry", March 2017.

[MS-NLMP] Microsoft Corporation, "[MS-NLMP]: NT LAN Manager (NTLM) Authentication Protocol", May 2014.

[RFC4757] Jaganathan, K., Zhu, L., and J. Brezak, "The RC4-HMAC Kerberos Encryption Types Used by Microsoft Windows", RFC 4757, DOI 10.17487/RFC4757, December 2006, <http://www.rfc-editor.org/info/rfc4757>.

[RFC6649] Hornquist Astrand, L. and T. Yu, "Deprecate DES, RC4-HMAC-EXP, and Other Weak Cryptographic Algorithms in Kerberos", BCP 179, RFC 6649, DOI 10.17487/RFC6649, July 2012, <http://www.rfc-editor.org/info/rfc6649>.

[RFC7465] Popov, A., "Prohibiting RC4 Cipher Suites", RFC 7465, DOI 10.17487/RFC7465, February 2015, <http://www.rfc-editor.org/info/rfc7465>.

Appendix A. Acknowledgements

Many people have contributed to the understanding of the weaknesses of these encryption types over the years, and they cannot all be named here.

Authors' Addresses

Benjamin Kaduk
Akamai Technologies

Email: kaduk@mit.edu

Michiko Short
Microsoft Corporation

Email: michikos@microsoft.com