Independent Submission
Internet-Draft
Updates: 5536,5537 (if approved)
Intended status: Standards Track
Expires: September 7, 2017

M. Baeuerle
STZ Elektronik

March 6, 2017

Cancel-Locks in Netnews articles
draft-baeuerle-netnews-cancel-lock-00

Abstract

This document defines an extension to the Netnews Article Format that may be used to authenticate the removal or replacement of existing articles. If approved, this document updates [RFC5536] and [RFC5537].

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 September 7, 2017.

Copyright Notice

Table of Contents

1. Introduction

The authentication system defined in this document is intended to be used as a simple method to verify that the author of an article which removes or replaces another one is either the poster, posting agent, moderator or injecting agent that processed the original article when it was in its proto-article form.

One property of this system is that it prevents tracking of individual users.

There are other authentication systems available with different properties. When everybody should be able to verify who the originator is (e.g. for control messages to add or remove newsgroups) an OpenPGP signature is suited.

1.1. Conventions Used in This Document

Any term not defined in this document has the same meaning as it does in [RFC5536] or [RFC5537].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2. Author's Note

Please write the letters "ae" in "Baeuerle" as an a-umlaut (U+00E4, "ä" in XML) and the letters "ue" in Baden-Wuerttemberg as an u-umlaut (U+00FC, "ü" in XML).

2. Header Fields

This section describes the formal syntax of the new header fields using ABNF [RFC5234]. It extends the syntax in Section 3 of [RFC5536] and non-terminals not defined in this document are defined there. The [RFC5536] ABNF should be imported first before attempting to validate these rules.

The new header fields Cancel-Lock and Cancel-Key are defined by this document:

```
fields =/ *( cancel-lock / cancel-key )
```

Each of these header fields MUST NOT occur more than once in an article.

Both new header fields contain lists of encoded values. Every entry is a <code-string> based on a <scheme>:

```
scheme       = %s"sha-256" / 1*20scheme-char / obs-scheme
scheme-char  = LOWER / DIGIT / "-"

code-string  = 1*base64-octet
base64-octet = ALPHA / DIGIT / "+" / "/" / "="
```

The hash algorithms for <scheme> are defined in [SHA], see also [RFC1321] and [RFC6151] for MD5, [RFC3174] for SHA1 and [RFC6234] for the SHA2 family. The Base64 encoding used is defined in Section 6.8 of [RFC2045].

This document defines one value for <scheme>: "sha-256". This value is mandatory to implement

Note that the obsolete syntax <obs-scheme> was defined case-insensitive. This is changed in this document and the scheme MUST now be generated with lowercase letters.

Values for <scheme> defined in future updates to this document are length limited to 20 characters. This and the case-sensitivity are defined to simplify the checks.

2.1. Cancel-Lock

```
cancel-lock  = "Cancel-Lock:" SP c-lock *(CFWS c-lock) [CFWS]
c-lock       = scheme ":" code-string
```

If <scheme> is not supported by an implementation, the corresponding <c-lock> element MUST be skipped and potential following <c-lock> elements MUST NOT be ignored.

The <code-string> in <c-lock> is the Base64 encoded output of a hash operation (defined by <scheme>) of the Base64 encoded key that is intended to authenticate the person or agent that created or processed respectively the article before injection. Because of the one-way nature of the hash operation the key is not revealed.

2.2. Cancel-Key

```
cancel-key   = "Cancel-Key:" SP c-key *(CFWS c-key) [CFWS]
c-key        = scheme ":" code-string
```

If <scheme> is not supported by an implementation, the corresponding <c-key> element MUST be skipped and potential following <c-key> elements MUST NOT be ignored.

The <code-string> in <c-key> is the Base64 encoded key that was used to create the Cancel-Lock header field as defined in Section 2.1 of the original article.

3. Use

3.1. Adding an initial Cancel-Lock header field to a proto-article

A Cancel-Lock header field MAY be added to a proto-article by the poster or posting agent which will include one or more <c-lock> elements.

If the poster or posting agent doesn't add a Cancel-Lock header field to an article, then an injecting-agent (or moderator) MAY add one provided that it positively authenticates the author. The injecting-agent (or moderator) MUST NOT add this header to an article unless it

is able to authenticate all remove or replace attempts from the poster and automatically add a working Cancel-Key header field for such articles.

Other agents MUST NOT add this header to articles or proto-articles that they process.

3.2. Extending the Cancel-Lock header field of a proto-article

If a Cancel-Lock header field has already been added to a proto-article then any agent (prior to the article being injected) further processing the proto-article (moderators and injecting-agents) MAY append a single <c-lock> element to those already in the header.

No more than one <c-lock> element SHOULD be added by each agent that processes the proto-article.

Once an article is injected then this header MUST NOT be altered. In particular, relaying agents beyond the injecting agent MUST NOT alter it.

3.3. Adding a Cancel-Key header field to a proto-article

The Cancel-Key header field MAY be added to a proto-article containing a Control or Supersedes header field by the poster or posting agent which will include one or more <c-key> elements. They will correspond to some or all of the <c-lock> elements in the article referenced by the Control (with "cancel" command as defined in [RFC5537]) or Supersedes header field.

If, as mentioned in Section 3.2 an injecting agent (or moderator) has added a Cancel-Lock header field to an article listed in the Control (with "cancel" command as defined in [RFC5537]) or Supersedes header field then (given that it authenticates the poster as being the same as the poster of the original article) it MUST add (or extend, if already present) the Cancel-Key header field with a <c-key> element that correspond to those article.

Other Agents MUST NOT alter this header.

3.4. Check a Cancel-Key header field

When a serving agent receives an article that attempts to remove or replace a previous article via Control (with a "cancel" command as defined in [RFC5537]) or Supersedes header field, the system defined in this document can be used for authentication. The general handling of articles containing such attempts as defined in [RFC5537] is not changed by this document.

To process the authentication, the received article must contain a Cancel-Key header field and the original article a Cancel-Lock header field. If this is not the case, the authentication is not possible (failed).

For the authentication check every supported <c-key> element from the received article is processed as follows:

1. The <code-string> part of the <c-key> element is hashed using the algorithm defined by its <scheme> part.

2. For all <c-lock> elements with the same <scheme> in the original article their <code-string> part is compared to the calculated hash.

3. If one is equal, the authentication is passed and the processing of further elements can be aborted.

4. If no match was found and there are no more <c-key> elements to process, the authentication failed.

4. Calculating the key data

This section is informative, not normative.

It is suggested to use the function HMAC(mid+sec) to create the key for an article with Message-ID <mid>, where HMAC is outlined in [RFC2104]. <sec> is a secret held locally that can be used for multiple articles. This method removes the need for a per-article database containing the keys used for every article.

The local secret <sec> should have a length of at least the output size of the hash function that is used by HMAC (32 octets for SHA-256). If the secret is not a random value, but e.g. some sort of human readable password, it should be much longer. In any case it is important that this secret can not be guessed.

Note that the hash algorithm used as base for the HMAC operation is not required to be the same as specified by <scheme>. An agent that verifies a Cancel-Key simply check whether it matches one of the Cancel-Locks.

Common libraries like OpenSSL can be used for the cryptographic operations.

5. Examples

Matching pair of Cancel-Lock and Cancel-Key header fields:

```
Cancel-Lock: sha-256:RrKLp7YCQc9T8HmgSbxwIDlnCDWsgy1awqtiDuhedRo=
Cancel-Key: sha-256:sSkDke97Dh78/d+Diu1i3dQ2Fp/EMK3xE2GfEqZlvK8=
```

Legacy variant:

```
Cancel-Lock: sha1:bNXHc6ohSmeHaRHHW56BIWZJt+4=
Cancel-Key: ShA1:aaaBBBcccDDDeeeFFF
```

Manual checks using the OpenSSL command line tools in a POSIX shell:

```
$ printf "%s" "sSkDke97Dh78/d+Diu1i3dQ2Fp/EMK3xE2GfEqZlvK8=" | openssl dgst
-sha256 -binary | openssl enc -base64
RrKLp7YCQc9T8HmgSbxwIDlnCDWsgy1awqtiDuhedRo=

$ printf "%s" "aaaBBBcccDDDeeeFFF" | openssl dgst -sha1 -binary | openssl
enc -base64
bNXHc6ohSmeHaRHHW56BIWZJt+4=
```

6. Obsolete Syntax

Implementations of earlier drafts of this specification allowed other <scheme> values and more liberal (case insensitive) syntax than is allowed in this version. The following values for <scheme> are now deprecated and SHOULD not be generated anymore. Serving agents SHOULD still accept them for a transition period as long as the corresponding hash function is not considered unsafe. See Section 7 for details.

```
obs-scheme   = "md5" / "sha1"
```

<obs-scheme> MUST be parsed case-insensitive.

7. Security Considerations

The important properties of the hash function used for <scheme> are the preimage and second preimage resistance. A successful preimage attack would reveal the real Cancel-Key that was used to create the Cancel-Lock of the original article. A successful second preimage attack would allow to create a new, different Cancel-Key that matches a Cancel-Lock too. Both cases would break the authentication system defined in this document.

Collision resistance of the hash function used for <scheme> is less important. Finding two Cancel-Keys that matches an arbitary Cancel-Lock is not helpful to break the authentication system defined in

this document (if a specific article is defined as target). Only collateral damage like arbitrary deletion or spam is possible.

Currently there are no known practicable preimage and second preimage attacks against the hash functions MD5 and SHA1. Therefore there is no hurry to replace them. The reasons why this document specify SHA-256 (aka SHA2-256) are:

- The last draft for the authentication system defined in this document is nearly two decades old. The client side implementations are moving forward extremly slowly too (newsreaders from the last millenium are still in heavy use). What is defined today should be strong enough for at least the next decades.

- The collision resistance of MD5 and SHA1 is already broken, therefore they are now obsolete for digital signatures as used in TLS. It is intended that an implementation of the authentication system defined in this document can share the same cryptographic library functions that are used for TLS.

- It is intended that the same hash function can be used for <scheme> and (as base) for the HMAC that is suggested in Section 4. See notes below for HMAC-MD5 and HMAC-SHA1.

- The SHA2 family of hash algorithms is widely supported by cryptographic libraries. In contrast, SHA3 is currently not supported by e.g. OpenSSL.

The operation HMAC(mid+sec) as suggested in Section 4 must be able to protect the local secret <sec>. The Message-ID <mid> is public (in the article header). An attacker who wants to steal/use a local secret only need to break this algorithm (regardless of <scheme>), because Cancel-Keys are explicitly published for every request to modify or delete existing articles.

Even if HMAC-MD5 and HMAC-SHA1 are not considered broken today, it is desired to have some more security margin here. Breaking <scheme> only allows to authenticate a single forged modify or delete request. With <sec> in hand it is possible to forge such requests for all articles that contain Cancel-Locks based on Cancel-Keys generated with this <sec> in the past.

8. IANA Considerations

The Hash Algorithm registry is maintained by IANA. The registry is available at <http://www.iana.org/assignments/hash-function-text-names/>.

9. References

9.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <http://www.rfc-editor.org/info/rfc2045>.

[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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5536] Murchison, K., Ed., Lindsey, C., and D. Kohn, "Netnews Article Format", RFC 5536, DOI 10.17487/RFC5536, November 2009, <http://www.rfc-editor.org/info/rfc5536>.

[RFC5537] Allbery, R., Ed. and C. Lindsey, "Netnews Architecture and Protocols", RFC 5537, DOI 10.17487/RFC5537, November 2009, <http://www.rfc-editor.org/info/rfc5537>.

[SHA] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS 180-4, DOI 10.6028/FIPS.180-4, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

9.2. Informative References

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, DOI 10.17487/RFC1321, April 1992, <http://www.rfc-editor.org/info/rfc1321>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <http://www.rfc-editor.org/info/rfc2104>.

[RFC3174] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, DOI 10.17487/RFC3174, September 2001, <http://www.rfc-editor.org/info/rfc3174>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <http://www.rfc-editor.org/info/rfc6151>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <http://www.rfc-editor.org/info/rfc6234>.

Appendix A. Acknowledgements

The author acknowledges the original author of the Cancel-Lock authentication system as documented in draft-ietf-usefor-cancel-lock: Simon Lyall.

He has written the original draft and former version <https://tools.ietf.org/html/draft-ietf-usefor-cancel-lock-01>. This document is mostly based on his work and was originally intended as revision 02. It must be renamed because the USEFOR IETF WG is now closed.

Appendix B. Document History (to be removed by RFC Editor before publication)

B.1. Changes since draft-ietf-usefor-cancel-lock-01

- Renamed document because the USEFOR IETF WG is now closed.
- Added more details how to check Cancel-Key header fields in Section 3.4.
- Added more details to Section 7.
- Added updated ABNF for Cancel-Lock and Cancel-Key header fields.
- Deprecated "md5" and "sha1" schemes.
- Added "sha-256" scheme.
- Reworded the abstract section and added references.
- Added note to other authentication systems to Section 1.
- Added command line check examples to Section 5.

B.2. Changes since draft-ietf-usefor-cancel-lock-00

- References to SHA-160 changed to SHA1
- "scheme" is now a case insensitive token and the number "1" has been changed to "sha1".
- Added some examples and fixed the section numbering.
- Updated 2nd paragraph on section 2.2 to make clear what exactly is being hashed and how.

- Changed paragraph 2 of 3.1 to discourage injection-agents from adding the header.

- Removed the Clue-string as this complicated the scheme without adding realistic functionality

- Moderators can now add these headers under the same conditions as injection-agents.

Author's Address

Michael Baeuerle
STZ Elektronik
Hofener Weg 33C
Remseck, Baden-Wuerttemberg 71686
Germany

Fax: +49 7146 999061
EMail: michael.baeuerle@stz-e.de