Groupe de travail Réseau
Request for Comments : 3230
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

J. Mogul, Compaq WRL
A. Van Hoff, Marimba
janvier 2002

Résumés d'instances dans HTTP

Résumé
HTTP/1.1 définit un en-tête Content-MD5 qui permet à un serveur d'inclure un résumé du corps de réponse. Cependant, ceci est défini spécifiquement pour couvrir le corps du message réel, et non le contenu du fichier complet (qui peut être assez différent, si la réponse est une gamme de contenu, ou utilise un codage des deltas). Aussi, le Content-MD5 est limité à un algorithme de résumé spécifique ; d'autres algorithmes, tels que SHA-1 (Norme de hachage sécurisé) peuvent être plus appropriés dans certaines circonstances. Finalement, HTTP/1.1 ne fournit pas de mécanisme explicite par lequel un client peut demander un résumé. Le présent document propose des extensions à HTTP qui résolvent ces problèmes.

Table des matières

1. Introduction

Bien que HTTP soit normalement mis en couche sur un protocole de transport fiable, tel que TCP, cela ne garantit pas un transport fiable des informations de l'envoyeur au receveur. Divers problèmes, y compris des erreur indétectées de transmission, des erreurs de programmation, la corruption de données mémorisées, et des interventions malveillantes peuvent causer des erreurs dans les informations transmises.

Une approche courante du problème de l'intégrité des données dans un protocole réseau ou un système réparti comme HTTP est l'utilisation de résumés, de sommes de contrôle, ou de valeurs de hachage. L'envoyeur calcule un résumé et l'envoie avec les données ; le receveur calcule un résumé des données reçues, puis vérifie l'intégrité de ces données en comparant les résumés.

Les sommes de contrôle sont utilisées pratiquement sur toutes les couches de la pile IP. Cependant, différents algorithmes

de résumé peuvent être utilisés à chaque couche, pour des raisons de coût de calcul, parce que la taille et la nature des données à protéger varie, et parce que les menaces possibles contre l'intégrité des données varient. Par exemple, Ethernet utilise un contrôle de redondance cyclique (CRC, *Cyclic Redundancy Check*). Le protocole IPv4 utilise une somme de contrôle de compléments à un sur l'en-tête IP (mais pas sur le reste du paquet). TCP utilise une somme de contrôle de complément à un sur l'en-tête TCP et les données, et inclut un "pseudo en-tête" pour détecter certaines sortes d'erreur de programmation.

HTTP/1.1 [RFC2616] comporte un mécanisme pour assurer l'intégrité du message, l'en-tête Contenu-MD5. Cet en-tête est en fait défini pour les messages conformes à MIME dans une spécification autonome [RFC1864]. Selon la spécification HTTP/1.1 :

"Le champ d'en-tête d'entité Contenu-MD5 [...] est un résumé MD5 du corps d'entité pour les besoins de la fourniture de la vérification de l'intégrité de bout en bout du message (MIC, *message integrity check*) du corps d'entité".

HTTP/1.1 a emprunté Contenu-MD5 au monde de MIME sur la base d'une analogie entre les messages MIME (par exemple, les messages électroniques) et les messages HTTP (demandes à, ou réponses d'un serveur HTTP).

Comme exposé plus en détails à la section 3, cette analogie entre les messages MIME et les messages HTTP a provoqué une certaine confusion. En particulier, alors qu'un message MIME est autonome, un message HTTP peut ne pas contenir la représentation complète de l'état actuel d'une ressource. (Plus précisément, une réponse HTTP peut ne pas contenir une "instance" complète ; voir à la section 3 la définition de ce terme.)

Il y a au moins deux situations où cette distinction pose problème :

1. Lorsque un serveur HTTP envoie une réponse 206 (Contenu partiel) comme défini dans HTTP/1.1. Le client peut se former une vision d'une instance (par exemple, un document HTML) en combinant une entrée d'antémémoire avec le contenu partiel qui est dans le message.

2. Lorsque un serveur HTTP utilise un "codage de delta", comme proposé dans un autre document [RFC3229]. Un codage de delta représente les changements entre l'instance actuelle d'une ressource et une instance précédente, et est un moyen efficace de réduire la bande passante exigée pour la mise à jour des antémémoires. Le client se forme sa propre idée d'une instance en appliquant le delta contenu dans le message à une des ses entrées d'antémémoire.

On inclut ces deux sortes de transformations dans une catégorie potentiellement plus large qu'on appelle "manipulations d'instance".

Dans chacun de ces cas, le serveur pourrait utiliser un en-tête Contenu-MD5 pour protéger l'intégrité du message de réponse. Cependant, comme la MIC dans un champ d'en-tête Contenu-MD5 ne s'applique qu'à l'entité qui est dans ce message, et non à l'instance entière en cours de ré-assemblage, elle ne peut pas protéger contre des erreurs dues à la corruption des données (par exemple, des entrées d'antémémoire) des erreurs de programmation (par exemple, une application impropre d'un contenu partiel ou d'un delta) de certaines attaques malveillantes [RFC3229], ou de la corruption de certains en-têtes HTTP dans le transit.

Donc, l'en-tête Contenu-MD5, bien qu'utile et suffisant dans de nombreux cas, n'est pas suffisant pour vérifier l'intégrité de l'instance dans toutes les utilisations de HTTP.

Le mécanisme d'authentification par résumé [RFC2617] fournit (en plus de ses autres objectifs) une fonction de résumé de message similaire à Contenu-MD5, sauf qu'il comporte certains champs d'en-tête. Comme Contenu-MD5, il couvre un message spécifique, et non une instance entière.

1.1 Autres limitations de HTTP/1.1

Les sommes de contrôle ne sont pas gratuites. Le calcul d'un résumé consomme des ressources de CPU, et peut ajouter de la latence à la génération d'un message. (Certains de ces coûts peuvent être évités par une mise en antémémoire attentive du côté de l'envoyeur, mais dans de nombreux cas, une telle antémémoire n'aura pas un ratio de touches utile.) Transmettre un résumé consomme de l'espace d'en-tête HTTP (et augmente donc la latence et les exigences de bande passante du réseau). Si le receveur du message n'a pas l'intention d'utiliser le résumé, pourquoi l'envoyeur du message devrait il gaspiller des ressources à le calculer et l'envoyer ?

L'en-tête Contenu-MD5 implique, bien sûr, l'utilisation de l'algorithme MD5 [RFC1321]. D'autres algorithmes peuvent cependant être plus appropriés à certains objets. Cela inclut l'algorithme SHA-1 [12] et divers algorithmes "de prise

d’empreintes digitales" [7]. HTTP ne fournit actuellement aucun soutien normalisé pour l’utilisation de ces algorithmes.

HTTP/1.1 suppose apparemment que le choix de générer un résumé appartient à l’envoyeur, et il ne fournit aucun mécanisme pour que le receveur indique si une somme de contrôle serait utile, ou quels algorithmes de somme de contrôle il comprendrait.

2 Objectifs

Les objectifs de cette proposition sont :
1. la couverture par le résumé des instances entières communiquées via HTTP,
2. la prise en charge de plusieurs algorithmes de résumé,
3. la négociation de l’utilisation des résumés.

Les objectifs n’incluent pas :

- la protection de l’intégrité de l’en-tête
  Le mécanisme de résumé décrit ici ne couvre que les corps des instances, et ne protège pas l’intégrité des "en-têtes d’entité" ou autres en-têtes de message associés.

- l’authentification
  Les mécanismes de résumé décrits ici ne sont pas destinés à prendre en charge l’authentification de la source d’un résumé ou d’un message ou d’une instance. Ces mécanismes ne sont donc pas une défense suffisante contre de nombreuses formes d’attaques malveillantes.

- la confidentialité
  Les mécanismes de résumé ne fournissent pas la protection de la confidentialité du message.

- l’autorisation
  Les mécanismes de résumé décrits ici ne sont pas destinés à prendre en charge l’autorisation ou d’autres formes de contrôle d’accès.

Le mécanisme d’authentification d’accès par résumé de la [RFC2617] peut fournir une certaine protection de l’intégrité pour certains en-têtes HTTP, et assure effectivement l’authentification.

3. Terminologie

HTTP/1.1 [RFC2616] définit les termes suivants :

ressource Objet ou service de données de réseau qui peut être identifié par un URI, comme défini au paragraphe 3.2. Les ressources peuvent être disponibles dans de multiples représentations (par exemple, plusieurs langages, formats de données, taille, résolutions) ou varier par d’autres aspects.

entité Ce sont les informations transférées comme charge utile d’une demande ou réponse. Une entité consiste en méta informations sous la forme de champs d’en-tête d’entité et en contenu sous la forme d’un corps d’entité, comme décrit à la section 7.

variante Une ressource peut avoir une, ou plus d’une représentation associée à tout instant. Chacune de ces représentations est appelée “variante”. L’utilisation de ce terme “variante” n’implique pas nécessairement que la ressource soit soumise à une négociation de son contenu.

La définition du dictionnaire pour "entité" est "quelque chose qui a une existence et un objectif ou réalité conceptuelle séparés et distincts" [8]. Malheureusement, la définition de "entité" dans HTTP/1.1 est similaire à celle utilisée dans MIME [RFC2045], fondée sur une analogie entièrement fausse entre MIME et HTTP.

Dans MIME, les messages de la messagerie électronique ont des existences distinctes et séparées. MIME définit une "entité" comme quelque chose qui se "réfère spécifiquement aux champs d’en-tête définis par MIME et au contenu d’un message ou d’une des parties du corps d’une entité multipartie".

Cependant, dans HTTP, un message de réponse à un GET n’a pas une existence distincte et séparée. Il décrit plutôt l’état actuel d’une ressource (ou d’une variante, sous réserve d’un ensemble de contraintes). La spécification HTTP/1.1 ne fournit

pas de terme pour décrire "la valeur qui serait retournée en réponse à une demande GET au moment présent pour la variante choisie de la ressource spécifiée". Cela conduit à une formulation étrange dans la spécification HTTP/1.1 aux endroits où ce concept est nécessaire.

Il est trop tard pour corriger les défaillances terminologiques de la spécification HTTP/1.1, de sorte qu'à la place on définira un nouveau terme, pour les besoins du présent document :

instance L'entité qui serait retournée dans une réponse d'état 200 à une demande GET, à l'instant présent, pour la variante choisie de la ressource spécifiée, avec l'application de zéro, un ou plusieurs codages de contenu, mais sans l'application d'aucune manipulation d'instance (voir ci-dessous) ou codage de transfert.

Il est pratique de penser à une étiquette d'entité, dans HTTP/1.1, comme étant associée à une instance, plutôt qu'à une entité. C'est-à-dire que pour une certaine ressource, deux messages de réponse différents peuvent inclure la même étiquette d'entité, mais deux instances différentes de la ressource ne devraient jamais être associées à la même (forte) étiquette d'entité.

On définit aussi ce nouveau terme :

manipulation d'instance C'est une opération sur une ou plusieurs instances qui peuvent résulter en ce qu'une instance soit convoyée du serveur au client en plusieurs parties, ou en plus d'un message de réponse. Par exemple, un choix de gamme ou un codage de delta. Les manipulations d'instances sont de bout en bout, et impliquent souvent l'utilisation d'une antémémoire chez le client.

4. Spécification

Dans la présente spécification, les mots clés "DOIT", "NE DOIT PAS", "DEVRAIT", "NE DEVRAIT PAS", et "PEUT" sont à interpréter comme décrit dans la [RFC2119].

4.1 Spécification des paramètres du protocole

4.1.1 Algorithmes de résumé

Les valeurs d'algorithme de résumé sont utilisées pour indiquer un calcul de résumé spécifique. Pour certains algorithmes, un ou plusieurs paramètres peuvent être fournis.

algorithme-de-résumé = jeton

Le BNF pour "paramètre" est celui utilisé dans la [RFC2616]. Toutes les valeurs d'algorithme de résumé sont insensibles à la casse.

L'autorité d'allocation des numéros de l'Internet (IANA, *Internet Assigned Numbers Authority*) agit comme registraire des valeurs d'algorithme de résumé. Initialement, le registre contient les jetons suivants :

MD5 C'est l'algorithme MD5, tel que spécifié dans la [RFC1321]. Le résultat de cet algorithme est codé en utilisant le codage base64 de la [RFC2049].

SHA C'est l'algorithme SHA-1 [12]. Le résultat de cet algorithme est codé en utilisant le codage base64 de la [RFC2049].

UNIXsum C'est l'algorithme calculé par la commande UNIX "sum", telle que définie par la "Single UNIX Specification, Version 2" [13]. Le résultat de cet algorithme est une chaîne ASCII de chiffres décimaux qui représente la somme de contrôle de 16 bits, qui est le premier mot du résultat de la commande UNIX "sum".

UNIXcksum C'est l'algorithme calculé par la commande UNIX "cksum", telle que définie par la "Single UNIX Specification, Version 2" [13]. Le résultat de cet algorithme est une chaîne ASCII de chiffres décimaux qui représente le contrôle de redondance cyclique à 32 bits, qui est le premier mot du résultat de la commande UNIX "cksum".

Si d'autres valeurs d'algorithme de résumé sont définies, le codage qui leur est associé DOIT être représenté comme une chaîne entre guillemets *(quoted string)* ou NE DOIT PAS inclure ";" ou "," dans le jeu de caractères utilisé pour le codage.

4.2 Résumés d'instance

Un résumé d'instance est la représentation du résultat d'un algorithme de résumé, avec l'indication de l'algorithme (et de tout paramètre) utilisé.

résumé-d'instance = algorithme-de-résumé "=" <résultat codé de résumé>

Le résumé est calculé sur la totalité de l'instance associée au message. L'instance est un instantané de la ressource avant l'application de toute manipulation d'instance ou de tout codage de transfert (voir la section 3). L'ordre des octets utilisé pour calculer le résumé est l'ordre de transmission des octets défini pour le type de contenu de l'instance.

Note : Le résumé est calculé avant l'application de toute manipulation d'instance. Si une gamme ou un codage de delta [RFC3229] est utilisé, le calcul du résumé après le calcul de la gamme ou du delta ne donnerait pas un résumé utile pour vérifier l'intégrité de l'instance réassemblée.

Le résultat du résumé codé utilise le format de codage défini pour l'algorithme de résumé particulier. Par exemple, si l'algorithme de résumé est "MD5", le codage est base64 ; si l'algorithme de résumé est "UNIXsum", le codage est une chaîne ASCII de chiffres décimaux.

Exemples :

MD5=HUXZLQLMuI/KZ5KDcJPcOA==
sha=thvDyvhfIqlvFe+A9MYgxAfm1q5=
UNIXsum=30637

4.3 Spécifications d'en-tête

Les en-têtes suivants sont définis.

4.3.1 Want-Digest

Le champ d'en-tête de message Want-Digest indique que l'envoyeur désire recevoir un résumé d'instance sur les messages associés à l'URI de demande.

Want-Digest = "Want-Digest" ":" n° (algorithme de résumé [";" "q" "=" qvalue])

Si un algorithme de résumé n'est pas accompagné d'une qvalue, il est traité comme si sa qvalue associée était 1.0.

L'envoyeur veut accepter un algorithme de résumé si et seulement si il figure dans un champ d'en-tête Want-Digest d'un message, et si sa qvalue est différente de zéro.

Si plusieurs valeurs acceptables d'algorithme de résumé sont données, l'algorithme de résumé préféré de l'envoyeur est celui (ou ceux) qui ont la plus forte qvalue.

Exemples :

Want-Digest: md5
Want-Digest: MD5;q=0,3, sha;q=1

4.3.2 Digest

Le champ d'en-tête de message Digest fournit un résumé de message de l'instance décrite par le message.

Digest = "Digest" ":" n°(résumé-d'instance)

L'instance décrite par un message peut être entièrement contenue dans le corps de message, partiellement contenue dans le corps de message, ou pas du tout contenue dans le corps de message. L'instance est spécifiée par l'URI de demande et tout valideur d'antémémoire contenu dans le message.

Un champ d'en-tête Digest PEUT contenir plusieurs valeurs de résumé d'instance. Cela pourrait être utile, par exemple, pour des réponses supposées résider dans des antémémoires partagées par des utilisateurs qui ont des navigateurs différents.

Un receveur PEUT ignorer tout ou partie des résumés d'instance dans un champ d'en-tête Digest.

Un envoyeur PEUT envoyer un résumé d'instance utilisant un algorithme de résumé sans savoir si le receveur prend en charge l'algorithme de résumé, ou même sans savoir si le receveur va l'ignorer.

Exemples :

```
Digest: md5=HUXZLQLMuI/KZ5KDcJPcOA==
Digest: SHA=thvDyvhfIqlvFe+A9MYgxAfm1q5=,unixsum=30637
```

5. Négociation de Content-MD5

HTTP/1.1 fournit un champ d'en-tête Content-MD5 *(Contenu-MD5)*, mais ne fournit aucun mécanisme pour demander son utilisation (ou sa non utilisation). Le champ d'en-tête Want-Digest défini dans le présent document fournit les bases d'un tel mécanisme.

D'abord, on ajoute à l'ensemble des valeurs d'algorithme de résumé (au paragraphe 4.1.1) le jeton "contentMD5", avec la disposition que cet algorithme de résumé NE DOIT PAS être utilisé dans un champ d'en-tête Digest.

La présence de l'algorithme de résumé "contentMD5" avec une qvalue différente de zéro dans un champ d'en-tête Want-Digest indique que l'envoyeur souhaite recevoir un en-tête Content-MD5 sur les messages associés à l'URI de demande.

La présence de l'algorithme de résumé "contentMD5" avec une qvalue de zéro dans un champ d'en-tête Want-Digest indique que l'envoyeur va ignorer les en-têtes Content-MD5 sur les messages associés à cet URI de demande.

6. Considérations relatives à l'IANA

L'autorité d'allocation des numéros de l'Internet (IANA, *Internet Assigned Numbers Authority*) administre l'espace de noms pour les valeurs d'algorithme de résumé. Les valeurs et leur signification doivent être documentées dans une RFC ou autre référence permanente, revue par des pairs et directement disponible, avec un niveau de détail suffisant pour permettre l'interopérabilité entre mises en œuvre indépendantes. Sous réserve de ces contraintes, l'allocation des noms est faite au premier qui le demande, (voir la [RFC2434]).

7. Considérations pour la sécurité

Le présent document spécifie un mécanisme de protection de l'intégrité des données qui protège les données des instances HTTP, mais pas les en-têtes d'entité HTTP, contre certaines formes de corruption accidentelle. Il est aussi utile pour détecter au moins une attaque en mystification [RFC3229]. Cependant, il n'est pas destiné à une protection générale contre l'altération malveillante des messages HTTP.

Le mécanisme d'authentification d'accès par résumé de HTTP de la [RFC2617] apporte une certaine protection contre l'altération malveillante.

8. Remerciements

On ne sait pas trop qui le premier a réalisé que le champ d'en-tête Content-MD5 n'est pas suffisant pour assurer la protection de l'intégrité des données lorsque on utilise des gammes ou des deltas.

Laurent Demailly pourrait bien avoir été le premier à suggérer un en-tête de somme de contrôle indépendant de l'algorithme pour HTTP [3]. Dave Raggett a suggéré d'utiliser le terme "résumé" à la place de "somme de contrôle" [14].

9. Références

[RFC1321] R. Rivest, "Algorithme de résumé de message MD5", avril 1992. *(Information)*

[RFC1864] J. Myers, M. Rose, "Champ d'en-tête Contenu-MD5", octobre 1995. *(D.S.)*

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par 2184, 2231, 5335.)*

[RFC2049] N. Freed, N. Borenstein, "Extensions multi-objets de la messagerie Internet (MIME) Partie cinq : critères de conformité et exemples", novembre 1996. *(Remplace* RFC1521, RFC1522, RFC1590*) (D.S.)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre, 1998. *(Rendue obsolète par la RFC 5226)*

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par 2817)*

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", juin 1999. *(DS.)*

[RFC3229] J. Mogul et autres, "Codage des delta dans HTTP", janvier 2002. *(P.S.)*

[3] Laurent Demailly. "Re: Revised Charter". http://www.ics.uci.edu/pub/ietf/http/hypermail/1995q4/0165.html

[7] Nevin Heintze. "Scalable Document Fingerprinting". Compte rendu du second atelier USENIX sur le commerce électronique, USENIX, Oakland, CA, novembre 1996, pp. 191-200. http://www.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html

[8] Merriam-Webster. "Webster's Seventh New Collegiate Dictionary". G. & C. Merriam Co., Springfield, MA, 1963.

[12] National Institute of Standards and Technology. "Secure Hash Standard". Publication FIPS 180-1, Ministère U.S. du Commerce, avril 1995. http://csrc.nist.gov/fips/fip180-1.txt

[13] The Open Group. "The Single UNIX Specification, Version 2 - 6 Vol Set for UNIX 98". Document numéro T912, The Open Group, février 1997.

[14] Dave Raggett. " Re: Revised Charter". http://www.ics.uci.edu/pub/ietf/http/hypermail/1995q4/0182.html

10. Adresse des auteurs

Jeffrey C. Mogul
Western Research Laboratory
Compaq Computer Corporation
250 University Avenue
Palo Alto, California, 94305, U.S.A.
mél : JeffMogul@acm.org
téléphone : 1 650 617 3304 (email preferred)

Arthur van Hoff
Marimba, Inc.
440 Clyde Avenue
Mountain View, CA 94043
mél : avh@marimba.com
téléphone : 1 (650) 930 5283

11. Déclaration complète de droits de reproduction

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.